Distributed Denial of Service Attack Detection in Wireless Sensor Networks

Thesis
by
Zubair A. Baig
for fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Monash University
January, 2008
This thesis is dedicated to my parents, wife and my son
Abstract

Wireless sensor networks have emerged as a significant source for the study and analysis of data from the environment. These networks are deployed in harsh and inaccessible environments with the purpose of monitoring their respective surroundings, and generating observed readings, for delivery to a centralised entity, for further data analysis. Sensor nodes are tiny devices with limited available resources for performing all their sensory operations, and be sustained for their entire lifetime. Applications of wireless sensor networks such as battlefield monitoring, bushfire monitoring and surveillance, are mission-critical in nature. The timeliness and accuracy in the delivery of the sensory data affects several mitigation efforts that may be launched upon successful detection of a particular event in the environment. Therefore, it is essential to protect such networks from malicious attacks, that may be launched by the adversary-class, with the intent of causing loss to the network operations.

Distributed Denial of Service (DDoS) attacks are defined as attacks launched from multiple ends of a wireless sensor network towards a set of legitimate sensor nodes, with the intent of exhausting their limited energy resources. These attacks can significantly affect the performance of the network, and eventually lead to complete compromise of all sensor nodes of the network. The consequences of such an attack, if left undetected, can be catastrophic to the operations of the entire network.
In this thesis, we model distributed denial of service attack detection as a pattern recognition problem, and propose techniques for detecting such attacks. The topological nature of wireless sensor networks differentiates them from standard networks. We define specific topology-dependant patterns to model normal network traffic, to facilitate differentiation between legitimate traffic packets and anomalous attack traffic packets. We propose two attack detection techniques for various classes of adversaries, that may participate in the attack. These two techniques ascertain that the attack detection process is accomplished with minimal overhead in the presence of adversaries with varying capabilities. The two techniques rely on distributed pattern recognition for detection of such attacks. The distributed nature of the proposed algorithms ensures that most steps of the attack detection process are performed within the sensor network, without the need to communicate on a frequent basis with centralised network base stations. Several optimisation criteria, such as frequency of convergence of the detection scheme, and selection of specific detector and decision-making nodes, are addressed as part of the detection schemes to reduce the overhead incurred on the sensor resources.

We also perform an evaluation of the scheme through simulation experiments, to test the effectiveness of our approach. In addition, the quantitative results acquired from the experiments are benchmarked with corresponding results acquired from a centralised Self-Organising Map-based attack detection scheme. Through the result comparisons, we prove the significance of distributed pattern recognition in such networks, for detecting distributed denial of service attacks in a timely and energy-efficient manner.
List of Publications

Book Chapters


Journal Papers


Conference Papers


**Related Papers**


**Papers under review**


Other Contributions

Acknowledgments

I would like to acknowledge the moral and intellectual support given to me by my supervisors Dr. Asad Khan and Professor Bala Srinivasan during my PhD program. It was a very long, and at times, tedious journey, covered with smoothness, thanks to their constant guidance and approach. I would also like to thank my wife and my son for their moral support and patience during the course of my research work.

Zubair A. Baig

Monash University

January 2008
Distributed Denial of Service Attack Detection in Wireless Sensor Networks

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Zubair A. Baig
January 30, 2008
Contents

Abstract ........................................................................ iv

List of Publications ....................................................... vi

Acknowledgments ........................................................... ix

List of Tables ................................................................. xv

List of Figures ................................................................. xvii

1 Introduction ................................................................. 1

1.1 Security Requirements for Wireless Sensor Networks .......... 3

1.2 Distributed Denial of Service - The Internet ................. 5

1.3 Distributed Denial of Service - Wireless Sensor Networks ... 7

1.4 Motivation and Objectives ............................................ 9

1.5 Research Contribution .............................................. 11

1.6 Thesis Outline ...................................................... 13

2 Attack Models and Detection Techniques ....................... 16

2.1 Wireless Sensor Networks ......................................... 17

2.1.1 Terminology ................................................... 20
3.6 Conclusions .................................................. 94

4 Distributed Attack Detection Scheme ......................... 96
    4.1 Introduction .............................................. 98
        4.1.1 Preliminaries ........................................ 101
        4.1.2 Contributions ........................................ 102
    4.2 Attack Detection Scheme .................................. 103
        4.2.1 Phase 1: Initialisation .............................. 108
        4.2.2 Phase 2: Observation ............................... 116
        4.2.3 Phase 3: Communication ............................ 117
        4.2.4 Phase 4: Verdict .................................... 121
        4.2.5 Phase 5: Pattern Update ............................ 123
    4.3 Computation of the Optimal Time Epoch Length ($\Delta_{opt}$) .... 124
    4.4 Selection of the Decision-Making (mGN) Nodes .............. 130
    4.5 Efficiency Analysis ....................................... 136
    4.6 Conclusions ................................................. 138

5 Performance Analysis and Benchmarking ..................... 141
    5.1 Introduction ............................................... 142
    5.2 Analysis ................................................... 146
        5.2.1 Experimental Setup .................................. 146
        5.2.2 Energy Decay Rates .................................. 149
        5.2.3 Attack Detection Rates .............................. 157
        5.2.4 Pattern Update Rate .................................. 165
        5.2.5 False Alarm Rates .................................... 168
    5.3 Self-Organising Map-based Attack Detection ............... 172
        5.3.1 Learning Phase ...................................... 175
# List of Tables

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Specifications - The Berkeley Mica Mote Sensor Node</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Attack Comparison Table</td>
<td>41</td>
</tr>
<tr>
<td>2.3</td>
<td>Effectiveness and applicability of the proposed detection schemes to a wireless sensor network environment for purposes of distributed denial of service attack detection</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>Threshold subpatterns for a set of two example target nodes, to be stored one each within the d detector nodes</td>
<td>90</td>
</tr>
<tr>
<td>4.1</td>
<td>Notations for the Distributed Attack Detection Scheme</td>
<td>104</td>
</tr>
<tr>
<td>4.2</td>
<td>Flat Topology - Threshold subpatterns for target nodes t₀, t₉, t₁₁ and t₁₅, for storage within the GN nodes with ID given by ID(n)</td>
<td>112</td>
</tr>
<tr>
<td>4.3</td>
<td>Cluster-Based Topology - Threshold subpatterns for target nodes CH₁ and CH₂, for storage within the GN nodes with ID given by ID(n)</td>
<td>114</td>
</tr>
<tr>
<td>4.4</td>
<td>Data Aggregation Topology - Threshold subpatterns for two data aggregation paths, with target nodes: t₄, t₅, t₉, t₁₃, t₁₅, t₂₄, t₂₇, t₄₃, for storage within the GN nodes with ID given by ID(n)</td>
<td>116</td>
</tr>
</tbody>
</table>
5.1 $\Delta_{opt}$ (seconds) values for variations in $\alpha$ and $TI_e$ .......................... 155
5.2 Error rates (%) in detection for varying pattern update frequencies. ............................ 167
5.3 Energy Decay Rates for the SOM-based centralised detection scheme. ............................... 185
5.4 Detection Rate Comparison - distributed detection and SOM-based schemes ......................... 187
5.5 False Alarm Rate Comparison - Distributed detection scheme and SOM-based schemes ............... 188
5.6 Energy Decay Rate ($\mu$J/sec) comparison between the distributed detection scheme and SOM-based detection scheme for $\alpha=0.95$ and $TI=500$. .................................................. 189

6.1 Notations for the Compromise-Tolerant Attack Detection. .......................... 199
6.2 Threshold (sub-pattern) values for target nodes $R_1$ and $R_2$. .......................... 202
6.3 Energy Utilisation Rates for Cluster-Heads($\mu$J/sec) ........................................ 223
6.4 Comparison of Total Cluster-Heads and Total mGN Nodes and corresponding Energy Decay Rates. .......................................................... 226
## List of Figures

1.1 Distributed Denial of Service traffic initiating from Zombie nodes on the Internet. ............................................. 7  
1.2 A High-level illustration of a Distributed Denial of Service Attack in a Wireless Sensor Network. .............................. 9  
2.1 Attack Relationship Diagram for Wireless Sensor Networks . . 40  
2.2 An Artificial Neural Network ..................................... 53  
2.3 The Graph Neuron Mapping Phase ................................. 61  
3.1 Distributed flooding attack model - Wireless Sensor Networks 74  
3.2 Flat Topology ......................................................... 81  
3.3 Cluster-based Topology ............................................. 82  
3.4 Data Aggregation Topology ....................................... 83  
3.5 Pattern vectors reconstituted for comparison with predefined threshold values. ..................................................... 92  
4.1 Multi-tiered Overlay for Distributed Attack Detection; Layer 1: GN nodes, Layer 2: mGN nodes and Layer 3: Base Station. 99  
4.2 Phases of the attack detection scheme. Phase 2-5 are executed in each time epoch $\Delta_i$ ........................................... 109
4.3 Flat Network Topology with the GN Array Overlay.

4.4 Cluster-based Network Topology with the GN Array Overlay.

4.5 Data Aggregation Network Topology with the GN Array Overlay.

5.1 GN Node Energy Utilisation Rate vs. Application Aspect Value ($\alpha$). The peak energy consumption rates in $\mu$J/sec ($\alpha = 0.1$) is $86$ for $N=2048$. The energy consumption rate of $17$ $\mu$J/sec is lowest for $\alpha=1.0$ and $N=128$.

5.2 mGN Node Energy Utilisation Rate vs. Application Aspect Value ($\alpha$). The peak energy consumption rates in $\mu$J/sec ($\alpha = 0.1$) is $352$ for $N=2048$. The energy consumption rate of $32.3$ $\mu$J/sec is lowest for $\alpha=1.0$ and $N=128$.

5.3 Number of mGN Nodes vs. Total Number of Nodes. The peak energy consumption rates in $\mu$J/sec ($\alpha = 0.1$) is $86$ for $N=2048$. The energy consumption rate of $17$ $\mu$J/sec is lowest for $\alpha=1.0$ and $N=128$.

5.4 Energy decay rate of detector (GN), mGN and target nodes for varying values of $\Delta_{opt}$ (seconds), $TI = 500, N=1024$.

5.5 Attack Detection Rate vs. Application Aspect Ratio ($\alpha$) for $TI = 500$. The peak detection rate ($\alpha = 1.0$) is $38\%$ for $N=128$, $65\%$ for $N=256$, $71\%$ for $N=512$, $84\%$ for $N=1024$ and $92\%$ for $N=2048$. The detection rate is lowest for $\alpha=0.1$: $10\%$ for $N=128$, $31\%$ for $N=256$, $47\%$ for $N=512$, $61\%$ for $N=1024$ and $86\%$ for $N=2048$. 

xviii
5.6 Attack Detection Rate vs. Detector Node Ratio for \( N = 128 \).
The peak detection rate is approximately 72% for low traffic intensity and \( n = 100\% \). For \( n < 10\% \), the detection rate is negligible for all traffic intensities.

5.7 Attack Detection Rate vs. Detector Node Ratio for \( N = 256 \).
The detection rate approaching 70% even with high traffic intensities (\( TI=500 \)), and fewer than 100% \( n \) nodes required to attain high detection rates.

5.8 Attack Detection Rate vs. Detector Node Ratio for \( N = 512 \).
Peak detection rate of nearly 90% for as few as 20% detector nodes in low traffic intensities.

5.9 Attack Detection Rate vs. Detector Node Ratio for \( N = 1024 \).
Peak detection rate of 93% for low traffic intensities. Even high values of \( TI \) yield a detection rate of above 80% for higher \( n \).

5.10 Attack Detection Rate vs. Detector Node Ratio for \( N = 2048 \).
Peak rate of 97% for low traffic intensities. Only 10-15% of detector nodes needed to achieve high detection rates.

5.11 Attack Detection Rate vs. Network Size (\( N \)), for \( TI=500 \).
Higher values of \( n \) yield higher detection rates. Larger node deployment densities essential if fewer detector nodes are to be selected, to sustain high attack detection rates.

5.12 False Positive Rate vs. Node Deployment Density (\( N \)) for varying Traffic Intensities

5.13 False Negative Rate vs. Node Deployment Density (\( N \)) for varying Traffic Intensities

5.14 SOM overlay on base station
5.15 Initial Attack Detection Rate vs. Network Types for varying traffic intensities. A peak value of 92% is achieved for $N=2048$ and $TI=50$. The lowest detection rate is for $N=128$ and $TI=500$.

5.16 Average Attack Detection Rate vs. Rate of Decline of Energy Content in the Target Nodes.

5.17 Initial False Positive Rate vs. Network Types for varying traffic intensities. A high false positive rate of nearly 14% is observed for $N=128$ and $TI=500$, whereas a very low false positive rate of approximately 2% is observed for $N=2048$ and $TI=50$.

5.18 Average False Positive Rate vs. Rate of Decline of Energy Content in the Target Nodes. $TI=500$ A peak false positive rate of 30% is observable for all $N$ values, when 10% of the target node’s energy content is depleted.

5.19 Initial False Negative Rate vs. Node Deployment Density ($N$) for varying traffic intensities. The highest false negative rate value observed is 30% for $N=128$ and $TI=500$, and the lowest value observed is 5% for $N=2048$ and $TI=50$.

5.20 Average False Negative Rate vs. Rate of Decline of Energy Content in the Target Nodes. $TI=500$ A peak false negative rate of 62% is observable for all $N$ values, when 13% of the target node’s energy content is depleted.

6.1 A cluster-based network with a set of malicious (Compromised) sensor nodes participating in the attack.

6.2 Square grid network with side = $a$ and number of clusters = $c_{opt}$.
6.3 Optimal Number of Clusters vs. Number of Nodes for varying values of $\gamma$. 206

6.4 Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=128$). A peak detection rate of 34% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=50\%$, the detection rate becomes negligible for all cluster sizes. 216

6.5 Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=256$). A peak detection rate of 80% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=50\%$, the detection rate becomes negligible for all cluster sizes. The detection rates for $c=c_{opt}$ and $c=0.5c_{opt}$ are comparable. 216

6.6 Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=512$). A peak detection rate of 85% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=67\%$, the detection rate becomes negligible for all cluster sizes. 217

6.7 Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=1024$). A peak detection rate of 94% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=70\%$, the detection rate becomes negligible for all cluster sizes. 218

6.8 Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=2048$). A peak detection rate of 97% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=70\%$, the detection rate becomes negligible for all cluster sizes. 219

6.9 Detection Rate vs. $N$ for $q=10\%$. A peak value of 80% is observable for $N=2048$. 219
6.10 False Positive Rate vs. Node Deployment Density $N$ for varying Node Compromise Ratio ($q$). A peak value of 32% is observable for $q=70\%$ and $N=128$. 

6.11 False Negative Rate vs. Node Deployment Density $N$ for varying Node Compromise Ratio ($q$). A peak value of 68% is observable for $q=70\%$ and $N=128$. 

6.12 Attack Detection Rate vs. Node Compromise Ratio ($q$) for varying Node Deployment Densities ($N$) and $c=1$. The peak detection rate is 45% for $N=2048$ and $q=1\%$. The detection rate reaches zero for all $N$, when 70% of the nodes are compromised. 

6.13 Attack Detection Rate vs. Node Compromise Ratio ($q$) for varying Node Deployment Densities ($N$) and $c=c_{opt}$. The peak detection rate is 96.5% for $N=2048$ and $q=1\%$. The detection rate is very low for all $N$, when 70% of the nodes are compromised.
Chapter 1

Introduction

Trends in contemporary computing have led to two essential classifications of modern-day computer systems, namely, high-performance grid systems, and resource constrained wireless devices. The capabilities and purposes of these two systems fall at contrasting ends of the spectrum. While at one end, high-performance grid computers empower cutting edge scientific research by providing the necessary computing and storage capabilities, at the other end, tiny devices with limited resources provide ubiquitous, pervasive and on-demand computing. At the same time, the latter also serve as a significant and essential source of data generation for rendering and subsequent storage on high-performance grid computers. One such class of networks of wireless resource-bound devices that has gained significant attention over the past few years has been the Wireless Sensor Network.

Wireless sensor networks consist of a collection of hundreds to thousands of tiny devices called sensors or sensor nodes. All architectures and standards proposed for such networks are bounded by considerations for their limited
resources. The limited on-board memory resources of such tiny devices restricts the size of applications, program codes and actual data that can be stored in their memory. The on-chip processing capability of the Berkeley Mica sensor (Perrig and Tygar, 2002), operating at 4MHz, is several orders of magnitude less than that of a standard desktop processor. Sensor nodes are generally supplied with power from batteries (8 mW for a Mica sensor node). Program codes and applications that demand large numbers of CPU cycles for execution may exhaust the limited energy of the sensor node much earlier than the anticipated lifetime of the node. It is thus evident that most applications and programs designed for high-performance computing devices cannot be accommodated unaltered into the small memory space of sensor nodes. All applications and programs designed for such resource-constrained devices must be light-weighted and compact in nature.

In general, sensor networks follow a rooted data delivery model (i.e. topology), with a computing device called the base station at the root of the network. The base station has several orders of magnitude more power and a longer lifetime, as compared to a standard sensor node (Perrig et al., 2001). Moreover, the base station has larger storage capacity, and hardware to facilitate higher data rates on the communication channel. Operations of the base station include: network initialization, information dissemination, node activation and revocation tasks, and for interfacing with other sensor networks.

Sensor nodes are generally deployed in harsh and inaccessible environments for monitoring and reporting of real-world events to the base station. Common applications of these networks include bush fire monitoring, building structure monitoring, battlefield monitoring and surveillance. Each sensor node is prone to a plethora of possible malicious attacks that may be
launched by the adversary class from either within or outside the network. Deployment of sensor nodes over a larger geographical area makes them even more vulnerable to any of these attacks (Chan and Perrig, 2003).

In order to ensure the smooth and uninterrupted network operation in mission-critical environments, it is essential to protect these networks from attacks that may be launched by the adversary class, with the intent of causing loss or damage to the network.

1.1 Security Requirements for Wireless Sensor Networks

Following are the basic requirements for provisioning security in wireless sensor networks:

- **Data Confidentiality**: Certain readings observed and generated by a sensor node can be classified as sensitive data, and therefore, must be protected from eavesdropping by rogue sensors and/or intruders. A standard approach to protect the confidentiality of sensory data is to encrypt it using a cryptographic key. The resource constrained nature of sensor nodes makes it a challenge to generate, store, and use cryptographic keys of any kind, asymmetric or symmetric (Perrig et al., 2001).

- **Data Authentication**: The authentication of messages exchanged between the sensor nodes is necessary to ensure protection against hoax messages that may be injected into the network by an adversary. Such
an attack may have catastrophic consequences considering the mission-
critical nature of sensor applications.

- **Data Integrity**: Data integrity ensures that the received data is not
  modified or tampered with on its way from the sender to the receiver.
  For instance, in a bush fire sensing network, an adversary may attempt
to alter sensor readings to trigger an alarm which otherwise would have
been initiated only for actual emergency scenarios.

- **Data Freshness**: An old set of messages i.e. sensor readings may be
  replayed by an adversary to mock a potential emergency in a normal
  situation. Therefore, it is essential to ensure the freshness of all data
  exchanged within the sensor networks.

- **Availability**: Sensor nodes deployed in un-trusted environments for
  carrying out critical operations must be able to survive the expected
  battery lifetimes. Premature exhaustion of the limited battery lives of
  sensor nodes may have a catastrophic effect on operations of the entire
  network. An adversary may attempt to launch an attack against valu-
  able resources in the sensor network to exhaust their energy resources,
  and cause the network to be disabled from continuing to operate and
  carry out its designated tasks pertaining to environment sensing and
detection.

Such an attack leads to denied access for the base station to sensory
data, that may be crucial for critical applications. Therefore, these
types of attacks are referred to as Denial of Service (DoS) attacks. The
DoS attack may or may not be launched from a single end point of the
network, wherein a single compromised node or a node belonging to an
adversary, repeatedly sends hoax requests to a legitimate target sensor node with the intent of exhausting its limited energy resources. On the contrary, an intelligent attacker may launch the attack from multiple ends of the network by compromising enough available resources to ensure high success in the attack process. The distributed nature of this attack is called a Distributed Denial of Service (DDoS) attack.

Several popular schemes such as the Standard Network Encryption Protocol (SNEP) (Perrig et al., 2001) and $\mu$TESLA (Micro Timed Efficient Stream Loss-tolerant Authentication) have been proposed in the literature (Perrig and Tygar, 2002) to satisfy the data authentication, freshness, and confidentiality requirements for provisioning security in wireless sensor networks. However, very little research has been done to address the issue of availability of sensor nodes under an attack.

1.2 Distributed Denial of Service - The Internet

Denial of service attacks are defined as attacks that are launched by a set of malicious entities towards a victim, with the aim of incapacitating it from providing further service to legitimate clients. The objectives of the attack are achieved by exploiting either system/protocol-level vulnerabilities, or by forcing the victim to undertake computationally intensive tasks, such as exponentiating large integers for applications such as Diffie-Hellman key exchanges (Baig, 2003).
On the contrary, distributed denial of service attacks are defined as flooding attacks, that do not rely on any particular network or system-level weaknesses. Rather they tend to exploit the asymmetry that exists between the network line rate and the victim’s processing capabilities. Distributed denial of service attacks are based on the principal: "Power of many is greater than power of few" (Mirlovic et al., 2004). Such attacks are launched subsequent to subversion and/or compromise of legitimate client machines of the network. These compromised machines then participate in the attack process, and await an instruction signal from a master node. The master node initiates the attack by first scanning through the network in search of vulnerable machines. The discovered vulnerabilities are then exploited by the master attacker to gain access to these vulnerable machines, and to infect them with attacker code. The vulnerable machines are thus compromised by the master attacker node, for participation in the attack process. Subsequently, a trigger signal from the master attacker node invokes the attacker code processes within each of the compromised machines. All compromised machines actively participate in the attack process, and generate a large number of hoax traffic packets to overwhelm a set of predefined victim nodes, incapacitating them from further service delivery.

The malicious nodes launch such an attack by amassing a large clan of hosts to simultaneously send useless packets towards the victim, leading to a flood of requests at the victim’s end (Figure 1.1). The intensity of the traffic is high enough to incapacitate the victim or its network. The master attacker node installs patches of the attack program on innocent agents (legitimate hosts) called zombies. This program triggers the launch of a simultaneous attack by the colluding adversaries, towards a defined victim node on the
network. As a result, the victim is flooded with requests coming in from all
directions at an enormously high magnitude.

![Diagram of Distributed Denial of Service traffic initiating from Zombie nodes on the Internet.](image)

Figure 1.1: Distributed Denial of Service traffic initiating from Zombie nodes on the Internet.

### 1.3 Distributed Denial of Service - Wireless Sensor Networks

Distributed denial of service attacks in sensor networks may be defined as
attacks which are launched by an adversary triggering multiple zombie or
compromised sensor nodes to send hoax requests to a target sensor node in
the network at very short time intervals. As a consequence, the target node is
overwhelmed with more number of requests than its maximum processing ca-
pacity, thus incapacitating it from providing any further service to its clients
(Wood and Stankovic, 2002)(Perrig et al., 2004). Such attacks may also rely
on the usage of laptop-class adversaries, i.e. adversaries with a few orders
of magnitude higher computational power than normal sensor nodes, with forged identities of legitimate sensor nodes operating in the network. More specifically, distributed denial of service attacks in a sensor network may lead to exhaustion of the limited energy resources of a target node, owing to the large inflow of requests towards it. Therefore, we also refer to a distributed denial of service attack in a sensor network as a distributed energy-exhaustion attack. It may be noted that distributed energy-exhaustion attacks in sensor networks are analogous to flooding attacks in high-performance networks wherein, an adversary triggers the generation of a flood of requests towards the victim node from several ends of the network with the intent of incapacitating it from providing additional service.

In (Wood and Stankovic, 2002), the authors have classified denial of service attacks at various layers of operation within a typical sensor network. Sensor network design must incorporate the level of damage an adversary may cause to the functionality of the network, as well as the failure tolerance levels of the network, to ensure a certain degree of robustness to node or route failure. Moreover, the asymmetry in the resources between the sensor network and the adversary must be considered prior to design of any security scheme. It may happen that a sensor network deployed in enemy territory is subverted or disrupted by an already existing wired network or power grid existing in the field.

As can be seen from Figure 1.2, a single victim node may be targeted with overwhelming number of incoming requests from multiple ends of the network. The attacker nodes can either be legitimate but compromised nodes operating in the network, or be a laptop-class adversary, i.e. an adversary
with higher capabilities, using forged identities to generate a large set of legitimate packets for overwhelming the victim node. It is assumed that no pre-hand information is available to allude towards critical (potential victims) nodes in the network. Therefore, an adversary must have observation capabilities for a certain period of time to identify on the critical nodes in the network. Intelligent set of adversaries will launch the distributed denial of service attacks from multiple ends of the network so as to avoid being detected by a detection module observing traffic flow from a single point of origin in the network.

![Illustration of a Distributed Denial of Service Attack in a Wireless Sensor Network.](image)

**Figure 1.2:** A High-level illustration of a Distributed Denial of Service Attack in a Wireless Sensor Network.

1.4 **Motivation and Objectives**

Wireless Sensor Networks face a myriad of threats that may lead to the loss of sensory resources required for crucial operations of the network. The availability of sensor nodes is under constant threat from DoS/DDoS attacks.
Distributed denial of service attacks do not subvert or tamper with the actual sensory information. Rather, they exploit the disparity which exists between the network line-rate and the victim’s limited resource availability. Detecting and defending against such attacks in sensor networks is non-trivial (Chan and Perrig, 2003). While mitigation approaches are post-attack measures, the scope of this thesis is focused on distributed denial of service attack detection algorithms. The detection of such attacks is the first step towards any counter-measures, including mitigation, that may be necessary for appeasing the effects of the attack upon their successful detection.

The untrusted environments of operation of wireless sensor networks accompanied with their resource-constrained nature necessitate the use of light-weighted mechanisms for detection of such attacks. The issue of distributed denial of service attack detection in wireless sensor networks remains unsolved, and all proposed solutions for such attacks in high performance networks, due to their resource demanding nature, are impractical for unaltered deployment on these resource constrained networks. The lack of a gateway, as a single point of entry into the network increases the vulnerability of such networks, and further complicates the attack detection process. This is due to the fact that such attacks cannot be successfully detected by a single attack detector node that may be deployed for purposes of attack detection, as will be elaborated upon in the subsequent chapters.

The aim of this thesis is to design light-weighted, in-network, distributed, and scalable mechanisms for detection of distributed denial of service attacks in wireless sensor networks. We compare different techniques for detection of such attacks in high-performance networks, and arrive at the conclusion
that the most appropriate approach in wireless sensor networks is to use in-network, collaborative and distributed pattern recognition. We model distributed denial of service attacks as a pattern recognition problem. The purpose of having a distributed mechanism is to reduce the overhead associated with frequent communications by the sensor nodes, with a centralised entity such as the base station.

The proposed techniques are scalable and efficient attack detection schemes that incur minimal overhead on the network. The attack is modeled as a distributed pattern recognition problem, and the attack detection process is accomplished by observing deviations in network traffic flow from the norm. Another goal of this thesis is to ascertain a high degree of accuracy in the attack detection process by defining various network and algorithmic parameters, such as the node deployment densities. The techniques proposed in the following chapters for detection of such attacks operate under varying sensor network application scenarios, under the threat of adversaries with varying capabilities.

1.5 Research Contribution

The main contributions of this thesis are:

- **Attack Pattern Modeling**
  - Network Model: The normal network traffic flow is modeled as a pattern i.e. we postulate that all legitimate network traffic flow follows a certain pattern, that needs to be adhered to, for smooth
operations of the network. In other words, threshold patterns are defined for modeling legitimate network traffic.

– Attack Model: We model distributed denial of service attacks against potential target nodes in wireless sensor networks based on the network traffic flow, the current energy contents of potential target nodes, and the criticality levels of sensor nodes.

– Pattern Generation Techniques: We define pattern generation criteria, as well as the pattern update rates, required for sustenance of satisfactory attack detection rates, based on the network model.

• Proposal and analysis of an efficient and distributed attack detection scheme

– Distributed Attack Detection: We propose a distributed pattern recognition approach, to facilitate collaborative information processing and pattern reconstruction, by a set of attack detector sensor nodes, for attack detection.

– Detector Decision-Making Nodes: We propose an algorithm for finding a minimal set of attack detector decision-making nodes, for purposes of attack decision making.

– Frequency of Scheme Convergence: We define a tradeoff equation to define the optimal frequency of scheme convergence, to reduce the overhead of detection, without compromising the success in attack detection.

– Analysis of Distributed Attack Detection Scheme: We perform an experimental analysis of the proposed distributed attack detection
scheme, for variations in the algorithmic and the network-level parameters. The analysis is done in terms of the attack detection rate, false alarm rates and the energy utilisation rates.

- Self Organising Map Comparison: We apply Self-organising maps as a type of neural network, for centralised detection of distributed denial of service attack patterns, and compare the quantitative outcomes of this approach, to the proposed distributed pattern recognition technique.

- Proposal and analysis of a compromise-tolerant mechanism for attack detection in cluster-based sensor network topologies
  
  - Scheme definition: We define theoretical bounds on the parameters of a cluster-overlay on a wireless sensor network, to achieve a desired level of failure-tolerance and success in attack detection, in the presence of compromised sensor nodes, participating in the distributed denial of service attack.
  
  - Performance Analysis: We perform simulations to analyse the proposed scheme, for variations in the algorithmic and network-level parameters.

1.6 Thesis Outline

In Chapter 2, we review various distributed denial of service attack detection schemes for high performance networks. We define a detailed analysis of the derivation of distributed denial of service attacks from other malicious attacks in wireless sensor networks. We elaborate on the limited applicability
of existing distributed denial of service attack detection techniques of high performance networks, to wireless sensor network environments. Finally, the effectiveness of distributed pattern recognition to the attack detection process is elaborated upon.

In Chapter 3, we classify wireless sensor network traffic flow, based on various network topologies. Subsequently, we model distributed denial of service attacks for the defined topologies, in the presence of attack detector nodes. The attack model itself is defined as a pattern recognition problem, hence emphasizing the need for distributed pattern recognition for purposes of attack detection. A detailed study of network traffic features that need to be analysed by attack detector nodes, for purposes of attack detection, is elaborated upon.

In Chapter 4, we describe a distributed pattern recognition-based distributed denial of service attack detection scheme. Several dependencies and algorithmic parameters are defined, and optimisation criteria are proposed for selection of detection decision-making nodes, as well as the frequency of convergence of the detection scheme. Finally, we perform a qualitative analysis of the proposed algorithm in the last section of the chapter.

In Chapter 5, we perform a detailed simulation analysis of the distributed attack detection scheme proposed in the previous chapter, for variations in both algorithmic as well as the network-level parameters. We also study Self-organising maps (SOMs) for detection of distributed denial of service attacks, and perform a comparison of the acquired results with corresponding results from simulations of the distributed detection scheme proposed in Chapter 4.

In Chapter 6, we define a fault-tolerant approach towards detecting distributed denial of service attacks in the presence of compromised sensor nodes.
A detailed simulation analysis is subsequently done to study the effectiveness of our approach.

Chapter 7 summarises the contributions made through this thesis, and elaborates on several approaches that need to be taken for continuing with further research in this area.
Chapter 2

Attack Models and Detection Techniques

With the ever-increasing deployment of wireless sensor networks for critical applications, there exists the added demand for securing these networks. The most common forms of attacks on modern day computing systems and networks are not directly applicable on wireless sensor network environments. The primary reasons for this lack of applicability are: a) Resource constrained nature of sensor nodes, b) Lack of a single entry point to the network, c) Non-triviality in targeting specific ‘critical’ sensor nodes and d) Inaccessibility of sensor nodes.

The need for detecting attacks in these networks must be preceded by clear and concrete definitions of attack models for such networks. The model of a sensor network attack, once defined, needs to be analysed, so as to help in development of counter-techniques. Attacks can be countered by attack detection, attack mitigation and attack prevention. The detection of an attack is the first step in attack defence, that needs to be taken, before any
attack mitigation techniques are actually applied. Similarly, attack prevention, although non-trivial in nature, is another approach towards protection of the network from malicious attacks.

In the first section of this chapter, we introduce the nature of wireless sensor networks, with emphasis on the limited capabilities of sensor nodes. In Section 2, we give an in-depth analysis of known attacks and defence techniques in wireless sensor networks. The third part of this chapter details on attacks against availability of resources in both high-performance networks as well as wireless sensor networks. The last part of this chapter discusses distributed denial of service attack detection techniques proposed in the literature for high-performance networks, and there potential applicability in wireless sensor network environments.

2.1 Wireless Sensor Networks

Advances in wireless technology coupled with comparative developments in embedded system technology have led to the development of tiny devices called sensors. Sensors or sensor nodes are inexpensive and low power devices, and are generally deployed in harsh environments such as battlefields and bushes. Once deployed, sensor nodes generally self-configure, and form a routing topology to facilitate communication of sensed data to the base station, or to other sensor nodes (Perrig et al., 2001). The wireless sensor network which results from this self-configuration, in recent times, has emerged as a very important resource for monitoring and detection of critical events in their operational environments. Sensor nodes are expected to become ever-cheaper in the near future. With the value of a typical sensor
network being determined by the flexibility in node deployment, their capabilities will not improve, but rather may be reduced, to fit the needs of contemporary applications. In addition, they are expected to maintain the same level of performance if not less, on smaller chip-sets i.e. reduced physical dimensions. The resulting networks encumber the process of facilitating security of any kind in them (Karlof and Wagner, 2002). Table 2.1 illustrates the technical specifications of the Berkeley Mica Mote sensor node (Perrig and Tygar, 2002).

<table>
<thead>
<tr>
<th></th>
<th>8-bit, 4MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Storage</td>
<td>8KBytes Flash Memory</td>
</tr>
<tr>
<td>Primary Storage</td>
<td>512 Bytes</td>
</tr>
<tr>
<td>Operating System</td>
<td>TinyOS</td>
</tr>
<tr>
<td>O/S Code Space</td>
<td>3.5 KBytes</td>
</tr>
<tr>
<td>Active Power</td>
<td>8 mW</td>
</tr>
</tbody>
</table>

Table 2.1: Specifications - The Berkeley Mica Mote Sensor Node

The fewer cycles of CPU processing per second (4MHz processor) on a sensor node, imply that several arithmetic and logical operations are unavailable. In addition, the communication on standard radio channel at 10 Kbps is also very slow (Perrig et al., 2001). The operating system on the Berkeley Mote-class sensors, TinyOS (Hill et al., 2000), consumes 8 KByte of instruction flash memory, which forms a significant proportion of the total available storage capacity. The remaining memory being available for storage of both the security overlay, as well as the application code (Perrig et al., 2001). Sensor nodes are capable of handling message broadcasts, including both transmission and reception. In addition, each node is capable of communicating with the base station and receiving messages addressed to it.
A typical sensor network consists of hundreds to thousands of low-power, low-performance sensor nodes. A sensor network in general terms is a heterogeneous collection of sensor nodes, with the possibility of some nodes having different functionality than the others (Karlof and Wagner, 2002). Unlike Mobile Ad Hoc Networks (MANETs), wherein nodes are mobile, and have high computational power, sensor nodes are generally static post-deployment within the environment. Moreover, MANETs follow a peer-to-peer communication paradigm (Park and Corson, 1997), wherein nodes communicate with each other using a single-hop mechanism. Wireless sensor networks on the other hand follow one of the following three common taxonomies for data delivery (Tilak et al., 2002):

- **One to many**: Base station broadcasts or multicasts a message (possibly a query) to several or all of sensor nodes in the network.

- **Many to one**: Sensor nodes convey their individual sensory readings to either the base station or another sensor node, called the data aggregation node, responsible for aggregating sensory readings.

- **Peer communication**: Sensor nodes exchange messages with other nodes at the same level of communication hierarchy, for purposes of coordination and control message exchange. Such communication may also be done for monitoring and intrusion detection, as will be explained in later sections.

Sensor networks have a rooted topology, with a base station operating as a central entity for control, coordination, and delivery operations of the network. In addition, several intermediary sensor nodes can also participate
in the routing process by aggregating localised sensory data, for delivery to the base station. The purpose of having such intermediary aggregation nodes is to reduce the effects of routing on the overall energy consumption rates of the network. As a consequence, longevity in the lifetime of the sensor network is attained.

The most significant resource of a sensor node is its power. A typical Berkeley Mica Mote sensor node if operating at full power, will function for only two full weeks (Karlof and Wagner, 2002). It is therefore imperative to have sensor nodes operate at minimal duty cycle, so as to facilitate extended lifetimes. Moreover, the harsh environments of deployment of sensors encumbers the task of detecting and replacing dead or inactive sensor nodes.

2.1.1 Terminology

Following is a list of commonly used terms in a sensor network environment:

- **High-Performance Networks**: Standard IP-based computer networks.
- **Node**: A sensor node operating as part of a wireless sensor network.
- **Base Station**: A centralised entity in a wireless sensor network, responsible for the initialisation of sensor nodes, generation of data acquisition requests to sensor nodes, and actual acquisition of sensory data.
- **Adversary-class**: The malicious class of nodes participating in an attack.
- **Attacker node**: A sensor node participating in an attack.
• **Laptop-class Adversary**: A malicious entity operating in the vicinity of the network, and having several orders of magnitude more resources than the sensor nodes.

• **Target**: A sensor node against whom an attack is launched.

• **IDS**: An Intrusion Detection System detects malicious penetration of packets and/or actual entities into the network.

### 2.1.2 Wireless Sensor Network Applications

Sensor networks are deployed for several purposes such as (Karlof and Wagner, 2002):

• **Emergency response**: Emergency response systems that may be monitoring the state of a concrete structure such as a high-rise building, or a bridge rely on readings from sensor nodes to confirm status of the structure. In case, the sensor readings depict deviation from normal readings, an emergency alarm may be triggered to ensure the safety of people and the structures.

• **Energy Management**: Several factors affect the behaviour of power resources in large metropolitan cities. These may include the outside temperature, load on individual transmission lines, moisture content in the air, wire temperatures etc. Sensors deployed at various points on a power system compromising of several thousand transmission lines would ensure that the load on the system is closely observed for any overloading that may take place, leading to the triggering of an alarm.
• **Medical Monitoring**: The bodies of patients may be closely monitored by a group of sensor nodes to study readings obtained from their bodies. The sensors may also be used to detect any deviations from normal behaviour in the body of the patient. Sensors may also automatically trigger the release of time-critical medicinal doses into the patient’s body upon observation of a particular known medical condition.

• **Logistics/Inventory Management**: Monitoring the movement of goods across countries or continents from the supplier to delivery points is being done by remote sensors. If a shopping centre runs out of a particular item, the remote sensors deployed in the item shelf would trigger a signal to the supplier indicating the need for delivery of a new consignment.

• **Battlefield Monitoring**: Rather than having human beings detect the status of certain critical factors during a battle such as the weather conditions, the number of troops in the opposition army, and their intensity of movement towards the target, remote sensors can be sprayed into the battlefield to detect and indicate conditions that may help better approximate these factors.

### 2.2 Attack Models in Wireless Sensor Networks

Sensor nodes operate in hostile environments such as battlefields and surveillance zones. The mission-critical nature of sensor network applications
implies that any compromise or loss of sensory resource due to a malicious attack launched by the adversary-class, can cause significant damage to the entire network. Sensor nodes deployed in a battlefield may have intelligent adversaries operating in their surroundings, intending to subvert, damage or hijack messages exchanged in the network. The compromise of a sensor node can lead to greater damage to the network. The resource challenged nature of environments of operation of sensor nodes largely differentiates them from other networks. All security solutions proposed for sensor networks need to operate with minimal energy usage, whilst securing the network. We classify sensor network attacks into three main categories:

- Identity Attacks
- Routing Attacks
- Network Intrusion

2.2.1 Identity Attacks

These attacks intend to steal the identities of legitimate nodes operating in the sensor network. The intent is to facilitate rogue node participation to either deny the base station access to sensor readings, or to tamper with node readings.

Sybil Attack

A Sybil attack is defined as an identity attack wherein malicious devices illegitimately take on multiple identities in the network (Newsome et al., 2004). The malicious device’s additional identities resulting from such an attack are
termed as Sybil nodes. Messages received by a Sybil node are in actuality received by the malicious device, and all messages transmitted by the Sybil nodes are actually sent by the malicious device. Another version of such an attack is when the Sybil nodes are inaccessible for direct communication by legitimate operating nodes of the network. In such scenarios, the malicious device will act as intermediary node, receive the messages, and pretend to forward them to the Sybil nodes.

The process of identity stealing to launch a Sybil attack, can be performed in one of two ways:

(i) Fabricated Identity: The attacker can create arbitrary Sybil identities by generating arbitrary random numbers as identifiers for the Sybil nodes.

(ii) Stolen Identity: The attacker initially identifies the identities of legitimate nodes of the network. Subsequently, the stolen legitimate identities are assigned to the attacker-generated Sybil nodes.

Once the Sybil nodes have been successfully created by the malicious device, the actual attack is launched in one of several ways. For peer-to-peer networks involving replication and storage of distributed data across the network, such an attack will entail towards the storage of data on Sybil nodes. A Sybil attack if launched against the routing topology of a network can have catastrophic consequences (Newsome et al., 2004). For instance, a multipath routing channel may in fact be going through multiple Sybil nodes representing a single malicious entity. A geographical approach to such an attack is when the attacker places multiple Sybil nodes at various locations of the network.
A defence mechanism against such an attack is *validation*, defined as a process of verifying that the identity given by a node is true, and is the only identity presented by its corresponding physical sensor node. A radio channel verification technique is proposed in (Newsome et al., 2004). For this technique to function correctly, it is assumed that all sensor nodes have a single radio communication channel for data transmission. Using this technique, a legitimate node \( c \) can verify the number of its neighbours which are Sybil nodes. Node \( c \) performs this operation by initially assigning each of its \( n \) neighbours a different radio channel to broadcast on. The node \( c \) subsequently chooses a channel randomly to listen on. If the neighbour to which the particular channel being listened to was assigned is legitimate, the message must be heard by \( c \). Given that \( s \) of the \( n \) neighbours of \( c \) are Sybil nodes, the probability that node \( c \) detects a Sybil neighbour by listening to a randomly selected channel on which there is no data transmission, is given by: \( \frac{s}{n} \). For \( r \) repetitions of this test, the probability of non-detection of Sybil nodes becomes \( \left( \frac{n-s}{s} \right)^r \).

Another proposed technique to validate node identity is to use shared secret keys. Using this technique, nodes can establish secure links to other nodes. Key pooling is a mechanism by which \( k \) random keys drawn from a pool of \( m \) keys are assigned to each node of the network. At network initialisation, if two nodes discover that they share a common key, a secure communication link is established between them. A malicious node intending to launch a Sybil attack will capture a set of nodes in the network, and create a pool of keys extracted from the captured nodes. However it must capture large enough numbers of nodes to be able to create Sybil identities. Neighbouring nodes validate the identity of the Sybil nodes by verifying that
the Sybil node has in possession the keys that it claims to have. If it is discovered that the Sybil node does not have in its possession one or more expected keys, the legitimate node can confirm the Sybil node’s false identity.

The damage that is incurred on the network as a consequence of a Sybil attack can be appeased by the above techniques at the cost of large scale key generation, distribution, and subsequent use for neighbour verification. The effects of a Sybil attack if left undetected, will lead to further attacks.

Node Replication Attacks

A node replication attack is defined as an attack wherein an adversary injects one or more nodes into the network with the same identity as an existing node. Unlike a Sybil attack, where a set of fictitious nodes are created by the adversary, node replication attacks involve physical insertion of rogue nodes into the network. This attack assumes that the adversary nodes have the capabilities for changing and subverting existing topological information in the network, such as route and trust in the network (Parno et al., 2005). The centralised approach towards detecting such an attack is to have every node generate and transmit a list of its neighbours and their claimed identities to the base station. The base station does the verification and subsequent revocation, if need be, of replicated nodes.

A randomised multicast mechanism (Parno et al., 2005) for detecting such attacks performs node replication detection by having each neighbour node of a location-declaring node to multicast a copy of the node location, confidentially, to a set of randomly selected witness nodes. Based on the birthday paradox (Cormen et al., 2001), for a network with \( n \) nodes, if each location produces \( \sqrt{n} \) witnesses, at least one collision will occur with high probability.
In other words, the probability of at least one of the witnesses receiving conflicting location claims (replicate) is high. For a network with 10,000 nodes, the estimated storage required on each node for the protocol to operate is 3,600B, which is nearly 91% of a Berkeley Mica node’s total memory (Perrig and Tygar, 2002). Therefore, the protocol becomes inefficient and less feasible for deployment.

A second approach proposed by the authors for detecting such attacks is the line-selected multicast technique, which is based on the premise that all nodes in the network act as routers. Therefore if a line is drawn between two nodes of the network, an intermediary node with two passing lines through it will successfully detect any conflicting location claims. In practical terms, when a node α’s neighbours send out a location claim to r witnesses, each of the nodes on the route store a copy of the location claim. The intermediary nodes on the route check the node location claims with their locally stored location claims previously received. If a conflict is found, a node revocation process is invoked. This approach gains a significant edge over the previous one in terms of the communication requirements, with the cost of communication being $O(n\sqrt{n})$, as compared to $O(n^2)$ for the previous approach. However, the imposed memory requirements on the sensor nodes still remain the same, and therefore, are a cause for concern.

2.2.2 Route-based Attacks

Directed diffusion is defined as a mechanism to facilitate data retrieval from sensor nodes. It is based on the principle of data-centric routing (Krishnamachari et al., 2002)(Akkaya and Younis, 2005)(Al-Karaki and Kamal, 2004), wherein the base station broadcasts a request for a particular data type into
the sensor network. The sensor nodes with current readings matching the request, respond by transmitting their readings back to the base station. Intermediary nodes in the request dissemination process forward the base station’s request to their neighbours. A dissemination tree rooted at the base station is thus formed and reinforced. Subsequently the desired sensory data generated by the nodes on the constituted path are transmitted to the base station (Intanagonwiwat et al., 2000)(Intanagonwiwat et al., 2003).

The route dissemination messages may be targeted by an adversary intending to misinform sensor nodes with route paths for the diffusion process. A more interesting attack against this routing scheme is for an adversary to suppress all sensory data flow in the network by spoofing a data flow path (Karlof and Wagner, 2002). Further, the attacker can tamper with the data flowing through her, and also resort to selective forwarding to hinder smooth and correct functioning of the network.

A specific form of such an attack is known as the Sinkhole Attack (Karlof and Wagner, 2002), wherein the attacker lures sensor nodes to believe it to be a centralised node for aggregating all received sensor data, or in worse cases, even a base station. An interesting approach towards launching such an attack is for the attacker to advertise good data delivery paths towards the actual sink i.e. the base station. The resulting traffic is diverted through the attacker node(s), and is susceptible to dropping or tampering.

A Wormhole Attack (Hu et al., 2002) is launched by a pair of colluding adversary nodes by diverting traffic from one end of the network to another through an adversary communication channel formed between the rogue node pair. A direct consequence of such an attack is denied access to sensory data for the base station (DoS).
Rogue nodes on a routing path from a source to the base station may attempt to tamper with or discard legitimate data packets. One solution to such a problem is to have multiple routing paths between each source-destination pair so as to ensure reliability in packet delivery, such as the one given in (Deng et al., 2004), although the purpose of their proposed scheme was to tolerate attacks against the base station, the scheme is applicable to sensor-base station route protection as well.

Another form of attack known as the Homing Attack (Wood and Stankovic, 2002), involves the study of network traffic flow by an adversary with the intent of targeting the critical nodes in the network for a potential attack. A solution to this problem is to have all packet headers encrypted with shared keys between the legitimate sensor nodes. However, the compromise of a sensor node may allow the adversary to retrieve its secret keys and participate and study the flow of network traffic.

In a Black Hole Attack (Wood and Stankovic, 2002), nodes advertise zero-cost routes to every other node, forming routing black holes within the network. As their advertisement propagates, the network routes more traffic in their direction. In addition to disrupting message delivery, this causes intense resource contention around the malicious node as neighbors compete for limited bandwidth. These neighbors may themselves be exhausted prematurely, causing a hole or partition in the network. Authorized exchange of routing information is one solution for ensuring protection against black-holes, and misdirection. Monitoring of neighbouring nodes for correct routing behaviour is another solution to the DoS attack problem at the network layer.
2.2.3 Network Intrusion

Network intrusion is defined as unauthorised access to a system by either an external perpetrator, or by an insider with lesser privileges (Anderson, 1980) (Kumar, 1995) (Sundaram, 1996) (Vigna and Kemmerer, 1999). An attacker will intrude a system or a network with malicious intent, and attempt to cause damage. In addition, an intrusion may also result from unauthorised but non-malevolent activity by less-privileged but legitimate users of the system. Intrusion detection (Mukherjee et al., 1994) is the process of detection of all such unauthorised access. An intrusion detection system is defined as a system responsible for monitoring and detection of all such network or system-level intrusions (Sun, 2004).

In wireline networks, the intrusion detection process can be classified into two main categories, namely, Anomaly Detection and Misuse Detection. Anomaly detection is defined as the process of detecting deviations in network or system activity from a known normal behaviour profile. A pre-requisite for this approach is to train the intrusion detection system with normal network traffic behaviour patterns. Anomaly detection systems can detect both known as well as unknown attacks, as an attack in general, is defined as a deviation from normal baseline activity. However, anomaly-based detection systems have a significant likelihood of suffering from high false alarm rates, and may require extensive training with large scale datasets of normal network or system activity. Two most common approaches for anomaly detection are (Sun, 2004):

1. Statistical Analysis: A statistical profile of normal activity is built using historic data. Examples of data used to build such a profile include:
number of system access requests, type of activity, and time of activity. The resulting analysis of the observed activity flags an intrusion upon observation of deviations from the statistical profile of normal behaviour.

2. **Neural Networks**: The neural network is initially trained with a training data set depicting normal network or system activity. Subsequently, the trained neural network is introduced with the observed activity, for classification purposes.

Misuse detection on the other hand, is a technique that compares the network or system activity with a known set of signatures depicting malicious behaviour. Misuse detection systems have lower false alarm rates as compared to anomaly detection systems. Moreover, the outcome of the analysis depicts the true attack, if a signature match takes place. As a result, complete information on the type of attack taking place, becomes perceivable. Signatures of attacks need to be known beforehand so as to facilitate the misuse detection process. Computer viruses are detected based on this approach. A major disadvantage of misuse detection systems is the need for keeping the database of attack signatures up-to-date. Misuse detection generally use pattern recognition for detecting signatures of malicious activity. System or network activity is mapped as a pattern, for subsequent comparison with known patterns of malicious behaviour. Any match is flagged as an intrusion. Other approaches include expert systems (Lunt et al., 1988)(Javitz and Valdes, 1991), which use rule definitions of attacks for comparison and detection purposes.

In wireless environments, it is imperative to have a distributed mechanism in place for detecting intrusions. Few research efforts have been put
for intrusion detection in wireless sensor networks. The wireless environment of operation coupled with the limited resource availability, makes the task of designing intrusion detection systems for sensor networks, a daunting one. Unlike wired networks, wherein a limited set of IDS nodes for a fixed number of entry-points to the network suffice, in wireless networks, an attack as a consequence of a network intrusion can come from all directions of the network (Zhang and Lee, 2000). One of the few papers discussing WSN intrusion detection is (Anjum et al., 2004), wherein the authors have proposed a technique for optimal placement of tamper-resistant intrusion detection modules in wireless sensor networks. They define a minimum-sized cutset for the network such that at least a single intrusion detection node lies on each of the routing paths of the network. The scheme is effective against attacks launched along autonomous routes of the network. The scheme will be less effective against more advanced attacks that are launched by a set of colluding adversaries.

2.2.4 Miscellaneous Attacks

Physical layer Attacks

A Jamming attack (Wood and Stankovic, 2002) against a sensor node is defined as a physical layer attack, wherein the radio frequencies of the victim node are disrupted. A node can observe the constant energy of its neighbours to conclude on a jamming attack as opposed to node failure. The standard defence against jamming involves various forms of spread-spectrum communication techniques. If the adversary can permanently jam the entire network, effective and complete denial of service is achieved. An alternate but costly
strategy towards protection against such an attack is to use any available alternate modes of communication, such as infrared or optical, if the attacker has not jammed them as well.

**MAC layer Attacks**

An adversary may induce changes in the message transmission frame to nullify the authenticity of a complete data packet due to a checksum mismatch (Wood and Stankovic, 2002). The adversary may induce more errors than the maximum error checking abilities of the system. Repeated re-transmissions of packets due to collision misinformation from adversaries will cause the exhaustion of the battery power of the sensor nodes. Time division multiplexing, allowing individual sensor nodes to transmit packets only within their respective frames of operation reduces this problem to a certain extent, although collisions still exist here as well. Such an attack will deny access to sensory readings by the base station.

**Attacks against the Base Station**

The base station is central to all activity of the sensor network. Therefore, it is imperative to protect it from attacks that are intended to isolate and/or incapacitate the base station from participation in the activities of the network. Traffic analysis attacks launched by the adversary class against the sensor base station is done in one of three ways: a) flooding of hoax requests to the base station, b) remote spoofing of the base station for traffic misdirection (a.k.a. sinkhole attacks), and c) message eavesdropping to locate, and subsequently jam or destroy the base station. An inaccessible base station denies service to sensor nodes, and therefore traffic analysis attacks against
a base station may be classified as a Denial of Service attack in a sensor network.

Several approaches have been proposed in the literature to thwart such attacks. A multi-base station, redundant path setup mechanism is proposed in (Deng et al., 2004), so as to facilitate tolerance to failure of single base stations. The scheme assumes that messages are routed on several paths from the source node to different base stations, and therefore multiple copies of messages are stored in multiple base stations at any given time. The vulnerability of the multi-base station setup phase to spoofing attacks is countered by having a one-way hash function applied to all base station-generated messages. One-way hashes are initially defined up to the \( n \)th place by the base station \((h_n, h_{n-1}, \ldots h_0)\), and are then revealed in the reverse order, i.e. \( h_0 \) is revealed first. Any hash value in the sequence is verifiable by the previously revealed hash values. For instance, the second hash value in the chain \( h_1 \) is equal to \( f(h_0) \). A sensor node upon receiving a multi-hop setup message from the base station, verifies the message-origin authenticity by comparing the hash value with the outcome of the one-way hash verification process. Such an approach helps protect the scheme against spoofing attacks.

2.2.5 Denial of Service - Wireless Sensor Networks

A Denial of Service (DoS) attack is a form of an attack which attempts to reduce or zero-out the operational capabilities of the victim (Moore et al., 2001)(Gligor, 1984)(Baig, 2003). The victim of such an attack can either be a single node, a set of nodes, the base station, or even the entire network, in which case, true denial of service is experienced by the back-end sensor data storage resources, or end-users. Attackers either exploit weaknesses in
the system, for which patches are later issued upon discovery of the attack, or the victim is forced to undertake computationally intensive tasks, such as exponentiation with large integers for Diffie-Hellman key exchanges. Generally, DoS attacks launched by the adversary-class exploit bugs in the software, however, other potential causes of such attacks include programming errors, resource exhaustion and physical damage caused by environmental hazards.

The purpose of denial of service attack detection is to ensure that the damage caused by the attack to the network resources is minimised, by reducing the impact zone of the attack. Attack detection mechanisms must be efficient, and operate in real-time, to facilitate timely and accurate detection of denial of service attacks. The ability of the detection scheme to distinguish between an attack and legitimate traffic, helps lower the false positive rate of the scheme, defined as the ratio of the number of legitimate packets classified as attack packets, and the total number of packets in the network.

The capabilities needed by an adversary to initiate a denial of service attack in a wireless sensor network are minimal. Denial of Service attacks in wireless sensor networks aim at diminishing and/or exhausting the limited battery power of the sensor nodes. If the adversary class includes laptop-class adversaries with higher processing and communication capabilities than standard sensor nodes, the outcome of such an attack can be disastrous for the entire sensor network. Very little research has been done in the area of Denial of Service attack detection and defence in wireless sensor networks. A detailed classification of possible denial of service attacks at various layers of operation has been elaborated upon in (Wood and Stankovic, 2002). Two other techniques proposed in the literature for detection of denial of service attacks in wireless sensor networks are as follows:
Spam Attacks

In (Sancak et al., 2004), the authors define spam attacks as attacks launched by a set of nodes called anti-nodes, injected into the sensor network by an adversary. The total number of anti-nodes, $a$, is much smaller than the actual network size $n$. The anti-nodes initiate a spam attack by generating frequent unsolicited dummy messages to their legitimate neighbour nodes of the sensor network. Considering the rooted topology of a sensor network, the amount of traffic accumulating at the nodes closer to the sink i.e. the base station, is much larger than that accumulating at the leaf nodes. Consequently, nodes up the tree hierarchy will exhaust sooner than other nodes.

The proposed detection strategy involves detection of faulty messages by the base station. Discrepancies in the readings of neighbour sensors are tagged as anomalous messages. In addition, large-scale messages generated by the same set of nodes are also classified as anomalous in nature. Upon successful detection of a spam attack, the base station transmits a request to all nodes in the vicinity of the anti-node to avoid relaying any unauthenticated messages. All subsequent messages are authenticated using a MAC computation operation. The proposed scheme scales effectively as the additional burden of authenticating messages is imposed only on the nodes immediately present within the vicinity of the anti-nodes. The scheme relies on neighbour readings for attack detection. It will be ineffective in detecting flooding attacks launched by a set of colluding adversaries in the network, which will require collaboration amongst the detecting nodes.
Practical Entropy Estimator

In (Kim et al., 2006), a practical entropy estimator is used to differentiate between various samples of data flow in the network. The entropy estimator is used by key management nodes of the network such as the sink, cluster head, or the base station, to compare the value for entropy of a sample of messages with another sample. The differences in the values help classify the traffic into attack or normal. Messages are assumed to carry a key space number along with some key information. Nodes up the network hierarchy can observe deviations in the entropy value computed when a large set of traffic is generated from nodes exhibiting the same key information.

2.2.6 Distributed Denial of Service - Wireless Sensor Networks

Distributed Denial of Service attacks do not exploit any particular vulnerability in the system, but rather exploit the asymmetry that exists between the network line-rate and the server processing rate (Elliot, 2000)(Gligor, 2003)(Chang, 2002). As part of a distributed denial of service attack, the adversary amasses a large clan of hosts, called zombies, to simultaneously send useless packets towards the victim, leading to a flood of requests at the victim’s end. The intensity of the traffic is high enough to incapacitate either the victim, or its network from further operations. The distributed denial of service attack process consists of two stages, namely, zombie initiation, and attack launch. During the zombie initiation process, the adversary compromises vulnerable nodes in a network, and installs on them attacker source code, possibly in the form of script. The code is written as such that it awaits
a 'trigger' call from the adversary to participate in the actual attack process, wherein all zombies generate a large set of useless packets towards a set of victim nodes in the network (Dietrich et al., 2000)(Peng, 2004). The zombie nodes may either exist in the same network as the victim, or be a part of another network. The attacker script may instruct the zombies to generate packets with randomly selected source addresses. The intent being to hide the identities of the zombie nodes.

In high-performance networks, distributed denial of service attacks can be classified into two categories:

(i) Direct Attacks: In a direct attack, the attacker arranges to send a large number of attack packets directly to the victim. SYN flooding is the most common attack case, in which TCP SYN packets are sent to the victim’s server port. The victim will respond by sending back a SYN-ACK response to the source address of the packet. Since the source address of the packet was spoofed, the victim will not receive the third message of the 3-way handshake required for connection establishment in TCP. Thus the number of half open connections at the victim’s end consume all the available memory, forcing the victim to deny service to subsequent clients (including legitimate clients).

(ii) Reflector Attacks: In a reflector attack, intermediate nodes (reflectors), are used as innocent attack launchers. The attacker sends packets with source addresses set to the victim’s address. Without realizing that the packets had spoofed source addresses, the reflectors send the response to the requests to the victim. As a result, the victim’s link is flooded with responses to reflected packets (Chang, 2002).
On certain occasions, it may happen by coincidence that a large number of legitimate packets are generated in a small time span, for transfer towards a certain set of destination nodes. Such a large influx of legitimate packets is referred to as a flash crowd (Jung et al., 2002). The process of distinctly identifying distributed denial of service attack traffic, and differentiating it from flash crowds is non-trivial. In wireless sensor networks, the taxonomy of the network, defined as the frequency of data delivery operations performed by the sensor nodes, is predefined and configured within each sensor node, at network initialisation time. The rate of delivery of data is generally constant, and therefore, rarely will a flash crowd of messages be generated for delivery by the nodes to the base station. Under an attack, the sensor nodes or the base station of a wireless sensor network are analogous to the server of an IP-based network, being a victim of a flooding-based attack.

The various attack models described in section 2.2 can culminate into distributed denial of service attacks, and vice versa. In Figure 2.1, a relationship between the various attacks in a sensor network is illustrated. The probabilities of the described attacks to culminate into a denial of service attacks, along with their need for having colluding adversaries or rogue nodes in the network are given in Table 2.2.

A typical distributed denial of service attack can be launched by a malicious entity by instigating a set of Sybil nodes to simultaneously generate malicious traffic packets towards a set of victim nodes on multiple routing paths. A successful Sybil attack is easily detectable by traffic packet validation, as defined earlier. However, when the attacker injects nodes into the network as part of a node injection or replication attack, and launches a distributed denial of service attack against target nodes in the network,
the resources of the target nodes will exhaust soon, and consequently, the attacker can steal the identities of these nodes, and reallocate them to the injected rogue nodes, initially operating as fictitious Sybil nodes. A network wherein the uniqueness of node identities is verified at regular time intervals, the probability of detecting Sybil attacks is diminished in the event of distributed denial of service attacks. This is because legitimate neighbour nodes of a Sybil node will be flooded with large traffic inflow, incapacitating the total number of monitoring nodes of the network. The damage caused by a distributed denial of service attack in such scenarios is irreversible, and potentially catastrophic to all network operations.

Sensor network routes connecting the various sensor nodes and the base station in the form of a tree, can be affected by a large influx of traffic owing to distributed denial of service attacks. The limited bandwidth wireless channels will eventually drop legitimate packets traversing the network, owing to the large number of useless data packets generated and transmitted by the rogue nodes.

Compromised nodes can generate enormous amount of traffic in a short span of time towards a set of target nodes in the network. The net traffic
influx associated with the compromise of $i$ nodes in a network with $n$ nodes, where $i \ll n$, is aggravated in the event where all nodes further participate in a collusion-based flooding attack against critical sensory resources. Sensor networks operating without a mechanism in place to detect node replication attacks, will succumb to the large inflow of traffic flow towards the critical node set. The previously described techniques to detect node replication attacks have a reasonable degree of uncertainty in the detection process, as they rely on probabilistic assumptions for conducting the detection process, and are therefore not ideal for detecting distributed flooding attacks. Moreover, the overhead incurred by the proposed schemes in (Parno et al., 2005), make such schemes less practical for detection of collusion-based attacks, which will necessitate their extensions to collaboration and extensive communications.

It may be observed here that the success of the node replication attack is increased manifold if it results into a flooding attack. The resulting victim nodes can have their identities compromised by their rogue node replicas to ascertain a greater degree of damage to the entire network.

<table>
<thead>
<tr>
<th>Attack</th>
<th>DDoS Consequence Probability</th>
<th>Colluding Adversaries</th>
<th>Detection/Defence Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybil Node Replication</td>
<td>High</td>
<td>No</td>
<td>Probabilistic</td>
</tr>
<tr>
<td>Wormhole</td>
<td>High</td>
<td>No</td>
<td>Probabilistic</td>
</tr>
<tr>
<td>Network Intrusion</td>
<td>Med</td>
<td>Yes</td>
<td>Anti-jamming techniques</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>Yes</td>
<td>Tamper-resistant Nodes</td>
</tr>
<tr>
<td>Node Implant</td>
<td>High</td>
<td>No</td>
<td>Crypto-secrets</td>
</tr>
<tr>
<td>Node Compromise</td>
<td>High</td>
<td>No</td>
<td>Crypto-secrets/Validation</td>
</tr>
</tbody>
</table>

Table 2.2: Attack Comparison Table
In light of the resource-constrained nature of sensor nodes, accompanied with differences in the operational environment of the network, we stipulate that the problem of distributed denial of service attack detection in sensor networks must be solved using simple, distributed, in-network attack detection mechanisms.

2.3 Distributed Denial of Service Attack Detection in HPNs

We define High-Performance Networks (HPNs) as standard IP-based computer networks that consist of a set of client machines and servers. Both the clients as well as the servers are assumed to have enough capabilities, computation, memory and communication, to operate smoothly even under the presence of large volumes of traffic. The maximum processing rate of a server in an HPN is given by \( \frac{L}{\tau} = S \), during any time interval \( \tau \) or larger, where \( L \) is the queue length at the server, and \( S \) is the application server processing rate (requests/sec). HPNs are assumed to follow a network topology, wherein a front-end router or gateway is responsible for interfacing the client machine(s) with the outside network.

Fundamentally, the distributed denial of service problem is tackled using a three-tier approach: a) Attack prevention and preemption, b) Attack detection, and c) Attacker identification and post-attack mitigation. Following are attack detection techniques proposed for HPNs in the literature.
2.3.1 Agent-based Approach

In (Elliot, 2000), the author suggests using host-specific security agents, to ensure prevention of a local system from becoming a zombie agent, for participating unwittingly in a distributed denial of service attack. The proactive security agent automatically audits systems, continually finding problems, and fixing them. A centralised security agent must be deployed in an organisation, to regularly take fingerprints of the host machine. If any system changes have been made by the attacker on a host machine in the network, the auditor is authorised to either remove the zombie application affixed into the host, or to rollback the infected application code. Centralised monitoring of the scheme entails for frequent exchange of information in the network.

Considering a sensor network environment, the validation of sensor application code on individual sensor nodes of the network will require extensive communications on the wireless channel, at any time over longer distances to the base station. The very nature of such a detection algorithm can lead to the sensor network’s befall. Therefore, it can be concluded that the heavy resource usage incurred by the agent-based approach, makes this scheme inefficient for practical deployment on sensor nodes.

2.3.2 Active Shaping

In (Kashiwa et al., 2002), Kashiwa et al. suggest an active shaping-based approach for tackling the DDoS problem. In their method, program modules called Active Components (ACs) are loaded into the network nodes, which may be routers, to implement application-level functions to detect, backtrack, and defend against attacks at the network level. They propose an algorithm
for attack detection, which does analysis of traffic characteristics before taking any decisions. The AC observes the amount of traffic flowing in the network during a given time period, and if it exceeds the throughput threshold, it concludes that an attack is in progress, and creates suspicious signatures for the 'attack' packets.

The attack packets are classified either by the front-end router in the attacker’s network, by identifying malicious packets on the basis of spoofed source packet addresses, or by the local AC in the victim’s network, which observes unusually high traffic received from a select set of hosts on an access-control list. One of the drawbacks of this approach is the non-zero probability of legitimate packets being dropped. These packets may be arising from clients, who are unknowingly involved in a flash crowd of data packets at the server end, and thus may be denied service because of the false assumptions made by the AC.

The proposed scheme relies yet again on the presence of a gateway to the network, for traffic monitoring, analysis, and decision-making. Therefore, the scheme is not efficient for practical deployment in a wireless sensor network environment.

2.3.3 Anomaly Detection

Anomaly detection is an intrusion detection technique based on the premise that network intrusions by the adversary-class have a corresponding anomalous behaviour pattern depicted by the host or the network (Anderson, 1980) (Ghosh and Schwartzbard, 1999). The anomaly detection process consists of two steps: a) modeling and learning of baseline normal network traffic

44
behaviour, and b) observing any deviations in behaviour of network traffic from the baseline models. Anomaly detection systems compare current network traffic patterns with statistical models of past network or system behaviour. Significant deviations are flagged as potential intrusions, and subsequent action is taken. A DDoS attack is a consequence of network intrusion in HPNs. In such scenarios, the master attacker node places the DDoS attack code on strategically-placed legitimate nodes of the network, to await a trigger signal from the master attacker node. Network intrusion detection modules perform attack detection based upon observed deviations of network traffic flow. The observed unusual patterns of traffic behaviour are considered as potential flooding attack traffic. Subsequently, attack mitigation techniques are initiated to appease the effect of the DDoS attack, and necessary countermeasures are taken. Anomaly detection techniques use artificial neural networks to learn normal network behaviour, and classify observed traffic into normal or attack.

Covariance analysis is an anomaly detection technique for comparing and classifying the feature sets of observed traffic flow (Jin and Yeung, 2004a). For a system where \( p \) features are selected for observation, for each observed vector \( x_n \), during a time interval \( t \), a covariance matrix is generated, depicting the feature set values \( \{ f_1, f_2, ... f_p \} \) for vector \( x_n \). The difference between the mean of the covariance matrix, and the mean of the matrix is used as the parameter in classifying the observed traffic vectors. The normal traffic vectors will be distinct from the anomalous traffic vector. The scheme performs correlations between observed traffic by generating and storing large matrices in the attack detection system. Secondly, the correlation analysis experiment needs to be performed on a centralised entity, most probably being an access point to the
entire network, in HPNs. Therefore, it is less efficient in terms of both energy utilisation as well as memory usage, for practical deployment in a wireless sensor network environment.

2.3.4 Misuse Detection

Misuse detection is another intrusion detection technique, which defines and models specific attack patterns against a system. These generated patterns of potentially harmful network behaviour are stored in the intrusion detection modules of the network. The misuse detection process can be divided into two steps: a) Defining and generating patterns of network or system misuse, and b) Comparison of observed network traffic behaviour with stored misuse patterns. The success of the misuse detection process is heavily reliant on the knowledge of attack patterns possessed by the system designer. The higher the number of known patterns of network or system misuse, the higher the attack detection rate. Misuse detection systems generally suffer from very low false positive rates, as patterns of intrusive network traffic behaviour are known, and are compared with current network traffic, without a statistical dependance. On the contrary, the false negative rate of misuse detection systems can reach high rates in the presence of intruders whose attack patterns are novel, and unknown to the detection system.

Feature correlation techniques (Jin and Yeung, 2004b) (Morin and Debar, 2003) are a domain of misuse detection systems, which quantify the differences in the observed vectors of network or system behaviour from the covariance matrices stored in the memory of the detection system, depicting normal network/system behaviour. Features of network traffic are selected and translated into covariance matrices before the actual detection process.
is initiated. For a network with $S$ known attacks, a matrix $M_r$ is generated, depicting the differences in the covariance values of the various attacks. The resulting differences help classify the observed behaviour into one of $S$ categories of attacks. A drawback of such an approach is the need for beforehand knowledge of attacks, and constant regeneration of the covariance matrices to store novel attack instances. The overhead associated with regeneration of large-sized covariance matrices, and the need for knowledge of attacks beforehand, encumber the deployment of this technique in a wireless sensor network environment.

2.3.5 Ramp-up Behaviour Analysis

Traffic constituting a typical flooding-based attack is initiated from multiple zombie nodes, activated by a trigger message sent by a single master adversary node. The zombie nodes usually belong to several networks, and are not time synchronised. In (Hussain et al., 2003), the authors propose a technique for identifying and differentiating denial of service attacks launched from single attacker nodes, from flooding attack packets, which are generated by multiple zombie nodes. The detection process does a time-series analysis of the arriving traffic at the victim’s end, by studying the latencies of packet arrivals. Distributed denial of service attacks will display a ramped-up behaviour in latency observations of packet arrival, due to the unsynchronised activation of zombie nodes. On the contrary, denial of service attack packets will begin at full strength, and not show any ramped-up behaviour. As a result, the scheme does differentiation of single-source denial of service attack packets from multi-source distributed denial of service attacks. The lack of a single
entry point to a sensor network makes the process of time-series analysis of
traffic more difficult in these networks.

2.3.6 Attack Detection in Wireless Networks

The issue of distributed denial of service attack detection has been limitedly
addressed for wireless networks. In (Tan and Seah, 2005), the authors pro-
pose a filtering approach towards dropping attack traffic packets in a Mobile
AdHoc Network (MANET). A concise explanation of differentiating between
normal and attack traffic based on filters is given. The problems that remain
unaddressed include: where to install the traffic filter, how the traffic clas-
sification will take place, and the overhead incurred in terms of the energy
consumption rates and delays incurred. The extent of damage incurred by
such attacks remains very high, it may be noted that the actual exhaustion of
resources that is caused by such an attack is upperbound by the total amount
of resources available to the adversary class. Compromised or implanted sen-
sor nodes in the network will only cause damage to the target nodes until
their own energy resources sustain.

The wireless infrastructure has seen the emergence of tools such as the
SMSflooder (Sherriff, 2000), which launch flooding attacks against wireless
victims by installing and triggering distributed denial of service attack code
on zombie nodes belonging to the wireline medium. The success of the attack
is yet again dictated by the intensity of attack traffic the victim is inundated
with. A general approach towards mitigation of such attacks in the presence
of mobile nodes is given in (Geng et al., 2002). The proposed scheme operates
at two layers:
• Layer 1: Coordinated technological solutions, and

• Layer 2: Incentive structure.

Layer 1 has four sub-layers for serving the following purposes:

• Improving device security: tamper-resistant nodes to avoid being compromised by a master node, for serving as a zombie, as part of an attack.

• User-level traffic control: individual mobile devices can control the maximum number of request that they may receive in a finite period of time.

• Coordinated filters: to ensure coordination amongst the filters, that monitor various entry-points to the network.

• Trace-back: traffic to the origin, to shut down perpetrators of the attack at the source itself.

Layer 2 of the proposed architecture applies usage-based fees to the mobile nodes, so as to reduce unnecessary traffic generation and use of the communication channel by the zombie nodes.

In the presence of very few entry points to the network, the routers or gateways can be selected to be tamper-resistant. The lack of these entry points in a wireless sensor network environment make this task more difficult. The control of traffic inflow by each sensor node on an individual basis requires the storage and comparison of statistics of traffic inflow. It will also lead to the dropping of legitimate traffic packets, thus increasing the false positives of the network. User-level traffic inflow is not sufficient to decide on whether the traffic is anomalous or legitimate. The topological characteristics of a wireless sensor network impose the need for monitoring of traffic rates differently, for
each separate sensor node, depending on its topological placement in the network (Chapter 3). Therefore, the applicability of such an approach in a wireless sensor network is not practical in entirety, although the concept of coordination amongst attack detector nodes is applicable.

In Table 2.3, we illustrate the effectiveness of the various distributed denial of service attack detection techniques defined above, and define the shortcomings of the proposed approaches for application in a wireless sensor network environment. Apart from the policy-enforcement approach, all other techniques are not energy efficient, for practical deployment in a wireless sensor network environment. A coordinated flavour of policy-based attack detection may be implemented for distributed denial of service attack detection. However, the issue of collaboration and coordination for such a distributed technique is unaddressed in the literature.

### 2.3.7 Pattern Recognition for DDoS Detection

Flooding attack packets in a network can be represented as patterns of anomalous behaviour. The statistical features of observed traffic packets can be extracted and introduced to a trained neural network, for classification and attack detection purposes. The process of pattern recognition using artificial neural networks is an approach for detection of distributed denial of service attacks in HPNs. The detection schemes described in Section 2.3.2 cannot classify large datasets generated from observed traffic features accurately. If partial amount of data, depicting an attack or normal traffic behaviour, is initially introduced to a neural network for learning purposes; the neural network can subsequently recognise actual data with a certain degree of accuracy. Pattern recognition algorithms can perform the aforementioned tasks
<table>
<thead>
<tr>
<th>Detection Techniques</th>
<th>Effectiveness</th>
<th>Applicability to Sensor Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent-based</td>
<td>High Overhead, Centralised.</td>
<td>Non-trivial to analyse individual application code by base station (extensive communication).</td>
</tr>
<tr>
<td>Active-shaping</td>
<td>Front end gateway required, high false-positives.</td>
<td>Non-availability of front-end gateway.</td>
</tr>
<tr>
<td>Anomaly detection</td>
<td>High offline training costs, centralised.</td>
<td>Requires known statistics of network traffic flow, centralised solution.</td>
</tr>
<tr>
<td>Misuse detection</td>
<td>Prehand attack signatures required, centralised.</td>
<td>Attack signatures are non-trivial to be defined beforehand, centralised solution.</td>
</tr>
<tr>
<td>Rampup behaviour</td>
<td>Very effective approach, centralised.</td>
<td>Lack of a single gateway to the network encumbers its applicability.</td>
</tr>
<tr>
<td>Filtering technique</td>
<td>Filter installation issue unaddressed.</td>
<td>Not applicable in a distributed collaborative environment.</td>
</tr>
<tr>
<td>Policy-enforcement</td>
<td>Distributed, no-collaboration between nodes.</td>
<td>Coordination required for attack detection in sensor networks.</td>
</tr>
</tbody>
</table>

Table 2.3: Effectiveness and applicability of the proposed detection schemes to a wireless sensor network environment for purposes of distributed denial of service attack detection.
for learning and detecting patterns depicting attack and/or normal network behaviour. It may be noted that all the following proposed techniques require centralised processing, making them less applicable for efficient and time-bound operation in a wireless sensor network environment.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information (Abdi, 1994)(Lawrence, 1994). The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. An ANN is a collection of simple processing nodes, and a set of synapses, acting as communication links between them. The connections generally have an associated weight, which defines the bias of the network. A set of nodes in the ANN are tagged as input nodes, and another set as output nodes. The data introduced to the input nodes defines the set of output nodes to be triggered. A neural network thus performs a functional mapping from the input node set to the output node set.

Traditionally, ANNs have been used extensively for detecting network and system-level intrusions. Several such techniques based on various ANNs have been proposed in the literature.
Radial-basis Neural Network Approach

Radial-basis neural network functions (RBNNF) (Tsang et al., 2004) serve as a universal nonlinear approximator for detecting attacks otherwise undefined in the attack database, in a misuse detection system. Like most neural networks, the RBNNF has three layers of operation, namely, input, hidden and output. The traffic to be analysed is presented to the input layer of the RBNNF. The hidden layer is responsible for applying a nonlinear transformation from the input space to the hidden space. Synaptic weights connect neurons in the hidden layer to the output layer. The output of the network defined as the summation of the product of the synaptic weights \(w_{ji}\) and the hidden layer outcome \(\phi\), is given by:

\[
F(x) = \sum_{i=1}^{N} w_{ji} \phi_i, i = 1, ..., N \tag{2.1}
\]

\(\phi\) is a Gaussian function to perform the nonlinear mapping from the input space to the hidden space. Variations in the traffic flow, otherwise classified as normal traffic, can be detected using the RBNNF approximation technique. In addition to binary classification of traffic into normal or attack, multi-class
classification of the traffic into one of several categories of attacks is possible using this technique.

**Self Organising Maps**

A Self Organising Map (SOM) (Kohonen, 1995) is defined as a map of vector points in a two-dimensional plane, with each vector point defined as a distinct class of observed data. The topology of the map is preserved once learning is complete, and remains unchanged during the actual data classification process. SOMs have been proposed for intrusion detection systems, which are exposed to higher dimension data. The nodes (vector points) of a SOM are also known as neurons, and have associated weights. In (Bolzoni et al., 2006), a Self-organising map (SOM)-based ANN is used for classifying data packets into normal or anomalous. The SOM-based intrusion detection process can be divided into two phases:

- **Learning:** During this phase, the randomly generated SOM with random weights for each of its neurons, is introduced with input learning data. A winning neuron for a particular data packet is defined as the neuron with the closest weight in terms of Euclidean or Manhattan distance to that of the input data packet. A learning rate parameter, $\alpha$, is set for the SOM algorithm, which defines the rate at which the weights of neighbour neurons of a winning neuron, $\eta_i$, change. In addition, a radius parameter, $r$, defines the total number of neighbours, $\eta_i(n)$, whose weights will be updated if the neuron $\eta_i$ wins. The process of learning is iterative, and runs for a certain number of cycles. At the end of each cycle, the values of both $\alpha$ as well as $r$ are lowered. Consequently, at
the end of the learning cycle, a well structured map of neurons results, with nodes arranged in a two-dimensional space based on their weights.

- Classification: The actual data for classification is introduced to the SOM upon successful completion of all cycles of learning. The Manhattan or the Euclidean distance of the weight of the input data packet is compared with the weights of each of the neurons. The neuron with the closest weight to that of the input is tagged as the winner neuron for the particular data packet.

As a result, distinct classification of the input data is achieved, and the observer can study the classes assigned to the observed data, and decide on the intrusiveness level of the data packets. The accuracy of the ANN-based intrusion detection techniques strongly depends on the values selected for the parameters $\alpha$ and $r$ of the algorithm. The size of the map in a 2D space defines the total number of distinct classes of input. In addition, the features of the data packets, such as packet source address, size of the packet, packet type (TCP, UDP etc.), need to be selected before the learning phase of the algorithm is initiated.

Attack traffic generated as part of a flooding attack is anomalous in nature, due to its large magnitude, and therefore statistical anomaly detection techniques are appropriate for detecting such attacks. A statistical anomaly detection system is built using ANNs, to facilitate classification of observed traffic into normal or intrusive, after an initial learning process. During the learning phase, the ANN is trained with both normal and anomalous traffic samples, using randomly generated data. The ANN post-learning, is capable of classifying network traffic into normal or anomalous. Several techniques
for detection DDoS attack patterns based on statistical methods have been proposed.

**Adaptive Resonance Theory**

In (Jalili et al., 2005), the authors propose a two-tier approach for classifying attack traffic into flooding or normal. A statistical pre-processor is used to extract features of the observed traffic initially, followed by the actual classification of the input data based on an Adaptive Resonance Theory (ART)-based ANN. The scheme does extraction of observed traffic in small time intervals. All observed traffic is subsequently processed through the neural network engine at the end of the time interval, with the resulting output of the ANN defining the classification of the traffic into normal or anomalous. The output of the ANN is used for tagging a particular time interval as attack or normal.

Unlike SOMs, ART (Carpenter and Grossberg, 2003) neural networks do not require the total number of clusters (map size) to be defined by the user beforehand. Rather, the data clustering is performed at runtime based on the properties of stability and plasticity, i.e. total number of clusters of data are extendible while the algorithm is iterating, and at the same time, each cluster contains distinctly different data values as compared to other clusters. The ART network consists of three layers, namely, input, comparison and output. The first two layers have equal number of neurons, and the output layer has fewer neurons. The input layer is responsible for storing patterns, and each neuron of this layer has a one-to-one connection with corresponding neurons in the comparison layer.
The scheme proposed in (Jalili et al., 2005) operates by feeding in a set of binary input vectors extracted from the traffic into the ART network. A binary pattern, \( x \), presented to the input layer, is classified by the ART network into one of the existing categories of data values, based on its similarity index in the output layer, defined as (Kulakov and Davcev, 2005):

\[
T_i = \frac{|w_i \cap x|}{\beta + |w_i|}
\]  

(2.2)

\( \beta \) is a system parameter, and \( w_i \) is the weight of each neuron \( i \) of the ART network. The neuron with the highest \( T_i \) value for a given input pattern \( x \), is declared the winning neuron. A comparison operation is performed to test the similarity level of the winning neuron weight and the input pattern. If the weights are similar enough, the winning neuron is selected, and its weight is updated based on a learning rate parameter, \( \eta \). However, if the weight comparison yields a poor similarity index, the values of the output layer are reset, and another node in the output layer with closest \( T_i \) is selected. The main difference between SOM and ART networks is the online learning capability of the ART algorithm. Major disadvantages of SOM-based networks as compared to ART networks are the extensive requirement for off-line learning, and the fixed size of the maps, or output categories.

Summary

Self-organising maps require extensive off-line learning to improve their accuracies. Distributed denial of service attacks in wireless environments will require a collaborative and distributed detection mechanism. In addition, the necessity for constant update of the trained data set, based on the nature of
wireless sensor networks (Baig et al., 2006)(Baig and Khan, 2008), as will be illustrated in Chapter 3, makes the application of SOMs for attack detection in wireless sensor networks less practical. The limited memory and processing capabilities of sensor nodes implies that the SOM map size cannot scale beyond a certain size, as larger SOMs would entail for larger memory space, and more extensive processing of input patterns, and subsequent comparisons, for attack detection. As a result, the accuracy of the overall detection scheme is affected. The total number of weight comparison operations performed for each input pattern will increase linearly with increases in the map size, with 100 comparisons on an average taking place for a standard 10 x 10 SOM. ART networks exhibit added features of online learning, implying extensive online processing, and therefore added delays, and even more resource usage, as compared to SOMs. Moreover, an additional layer of processing is required for generating the actual outcome of the attack detection process in both SOMs as well as ARTs.

The overhead associated with the use of artificial neural networks is extended in scenarios where attacks launched by colluding adversaries need to be detected. Distributed denial of service attacks in wireless sensor networks are launched by colluding adversaries, and therefore any mechanism which relies on centralised processing of data sets is inefficient and will lead to high false alarm rates. For wireless sensor networks, it is therefore conjectured that a collaborative and distributed light-weighted mechanism is needed for distributed denial of service attack detection.
2.4 Distributed Pattern Recognition for Attack Detection

For large traffic flow, with the need for detecting signatures of normal or anomalous network traffic, pattern recognition is a very effective method, as can be seen from Section 2.3.7. However, pattern recognition algorithms such as SOM and ART are not sufficient for detecting distributed denial of service attacks in wireless sensor networks. The distributed nature of such attacks demands the need for a distributed mechanism in place for their detection.

By far, the only known algorithm for distributed pattern recognition in light-weight devices is the Graph Neuron (GN) (Khan and Mihailescu, 2004)(Khan, 2002). The graph neuron is a low-overhead, distributed pattern recognition algorithm, which uses graph-based representation of patterns, for rapid learning and efficient pattern recognition. The technique uses parallel in-network processing to circumvent the pattern-database scalability problem associated with graph-based methods (Tarjan and Trojanowski, 1977). Inherently, the GN is an associative memory approach towards pattern recognition. The earliest implementation of an associative memory system is the Hopfield network (Izhikevich, 1999). These networks have primarily been used for implementation of associative, or content-addressable memories, and a range of optimisation problems. Studies on Hopfield memory model show that the model is not scalable, and is limited by the number of processing and storage nodes in the network. From a sensor network perspective, increasing numbers of deployed nodes, if bearing a Hopfield network implementation on them, would perform inefficiently, and will be upperbound by the total number of nodes in the network. Compared to Hopfield networks, Back propagation
networks (BPNN) scale better, but at the cost of large overhead for learning new patterns (Nasution and Khan, 2008).

The GN algorithm overcomes the scalability limitations of Hopfield networks, and the large learning latencies of BPNNs. It is a finely distributed pattern recognition algorithm, which preserves the data relationships in a graph-like memory structure. The GN structure and layout is analogous to a directed graph, the processing nodes of the GN array are mapped as the vertex set $V$ of the graph, and the inter-node connections (i.e. the communication channels) belong to the set of edges, $E$. The communications are restricted to the adjacent nodes (of the array), hence there is no increase in communication overheads with corresponding increases in the number of nodes. The information presented to each of the nodes is in the form of a \{value, position\} pair. Each of these pairs represents a data point in a two-dimensional reference pattern space. Hence, the GN array converts complex spatial/temporal patterns into a graph-like structural representation, and then compares the edges of the graph with input patterns for memorisation of patterns, and their subsequent recall. The GN avoids increasing computational costs associated with increasing numbers of stored patterns, by spreading the computations over a finely grained network and limiting to communication to the nearest nodes - in-network processing.

The GN application stores new patterns and recalls previously encountered patterns by executing a fixed number of steps. A pattern is represented as a set of input pairs of position and value. These inputs are mapped onto a virtual array of processors by using the adjacency characteristics of the input e.g. alphabets and numbers would have their inherent adjacency characteristics; similarly images would have the frequency bands, intensity, and spatial
coordinates as the adjacency characteristics per pixel, and so on. For a reference pattern domain \( R \), the GN array represents all possible combinations of \( P \) in \( R \). In Figure 2.3, domain \( R = \{ X, O \} \), implying that the \( p \) sub-pattern values can have only one of two values at a given time. Hence, each GN node is initialised with a distinct pair \( p \in R \). Also, each GN node executes an instance of the full code associated with the GN algorithm, and therefore the computation overhead imposed on all nodes is the same. Each GN node maintains an updated list called the bias array in its local memory, which holds the position of the adjacent GN node to be contacted as a stimulus upon reception of a particular input subpattern. The bias array is initialised with the appropriate entries during the learning phase of the GN.

![Figure 2.3: The Graph Neuron Mapping Phase](image)

The learning phase of the GN algorithm may be categorised into three stages as shown in Figure 2.3:

(i) Mapping of input patterns: Input patterns in the form of \( p \) (value, position) pairs are sequentially broadcast through the network. The nodes based on their defined position values store the relevant input(s),
disregarding the remainder of the pattern. The input pattern length is assumed to be discrete. In other words, the GN does not perform recognition of continuous input patterns. From Figure 1, for pattern $P_1$, node 1 (N1) with position $= 1$, will store the first sub-pattern/pair of $P_1$ given as '$X$', and will ignore the rest of the message.

(ii) Synchronisation phase: A broadcast signal is sent out marking the end of the incoming pattern to all the nodes.

(iii) Bias array update: During this phase, each node contacts all of its adjacent nodes to acquaint itself with the sub-patterns learned by the neighbours. In addition, N1 will also store the position of the adjacent node that it will need to communicate with for this particular pattern. In this case, N1’s adjacent node is N5, for pattern $P_1$. It may be seen from Figure 1 that for the input pattern $P_1$ (XOX), node 1 will update its local bias array with the entry N5. Similarly, N5 will update its bias array with the entry N1, N3. Thus each bias array entry records the adjacent nodes being activated for a particular input pattern. A new pair is defined as the one which has a different set of adjacent GN nodes to the existing rows of the bias array. Stages 1 and 2 of the GN learning phase take place in a purely parallel manner. The GN algorithm continues with its process of recall of stored patterns, and memorisation of new input patterns, provided that further memorisations are permitted by the system.

The Graph Neuron algorithm is distributed in nature, and executes in parallel manner, and thus meets the requirement of a collaborative and distributed approach towards detection of distributed denial of service attacks in
wireless sensor networks. However, the original GN algorithm lacks a decision making infrastructure to allow direct implementation of the GN application on sensor nodes. In addition, the GN algorithm is proposed as a generic approach for pattern matching and detection. In this thesis, we redefine this technique, enhancing its steps of execution, to propose algorithms for distributed denial of service attack detection in wireless sensor networks.

2.5 Conclusions

The critical nature of applications of wireless sensor networks demand the need for their protection against malicious attacks that may be launched against sensory resources by the adversary-class. Several attacks such as: node compromise, node replication, node implant and Sybil, have been modeled and analysed in the literature. However, there exists a need for protecting the availability of sensor nodes under a malicious attack, so as to ensure that smooth operations of the network are minimally affected. Denial of service attacks in wireless sensor networks have been studied in the literature. However, as such no concrete attack detection and mitigation approach has been proposed yet. An improved version of an attack against the availability of sensory resources is the distributed denial of service attack. These attacks are launched from multiple ends of the network, and attempt to diminish the energy resources of legitimate sensor nodes. The ultimate purpose of such an attack is the replacement of the attacked sensor nodes with malicious nodes. These replacing nodes participate in the network operations with malevolence, and intend to mislead the base station with false or withheld sensory data.
In this chapter, we defined a high-level illustration of a distributed denial of service attack in a wireless sensor network. Further, we illustrated the derivation of such attacks from other known malicious attacks in such networks. Subsequently, we enlisted various attack detection techniques for distributed denial of service attacks in high-performance networks. The inability of the wireless sensor network to sustain the proposed techniques, due to the resource-constrained nature of sensor nodes, and the lack of a single entry point to the network, make all such techniques insufficient for as-is deployment. The adversary-class in a wireless sensor network may have varying capabilities, thus demanding the need for an energy-efficient and quick approach towards attack detection. The topological (data-delivery) models used in wireless sensor networks are different from standard high performance networks. These topological aspects must be incorporated into any proposed attack detection scheme, to achieve accuracy in attack detection.

In the following chapter, we define the standard network models for wireless sensor networks, as well as capability-based adversary-class. We further proceed to arrive at the finding that under the absence of known signatures of attacks, we can use distributed, threshold-based pattern recognition, to detect a flood of malicious network traffic in the network, from multiple-ends. Our findings also emphasize on the need for having multiple, collaborative attack detector nodes in the network, for achieving higher success in attack detection. In Chapter 3, the attack models and network traffic flow models under normal conditions, are defined. A pattern-based, collaborative and distributed attack detection scheme for wireless sensor networks is proposed in Chapter 4. A decision-making layer constitutes as part of the attack detection scheme, and consists of a set of decision-making sensor nodes, \(m\text{GN nodes}\).
that are selected based on our proposed mSelect algorithm. The proposed
detection scheme operates in several phases, that must be executed within
a certain fixed-length time frame, defined as a time epoch. We formulate an
equation to compute the length of a time epoch based on various algorithmic
and network parameters. A performance analysis of the proposed attack
detection scheme, under variations of the algorithmic and network-level pa-
rameters, is given in Chapter 5. We also signify the need for distributed
pattern recognition for attack detection, by quantifying the superiority of
our proposed scheme over an example centralised SOM-based approach. In
Chapter 6, we define an attack detection scheme, that is tolerant to a partic-
ular type of adversary-class, that operates with the purpose of compromising
sensor nodes, for launching an attack against other legitimate sensor nodes.
Chapter 3

DDoS Attack Pattern Modeling

The purpose of distributed denial of service attacks is to generate large volumes of traffic packets towards a set of victim nodes, thereby leading to rapid exhaustion of energy resources in them. Subsequently, the adversarial nodes replace the legitimate victim nodes, and participate in the network operations with malicious intent. These nodes will generate false and misleading sensory readings for delivery to the base station. Considering the critical nature of application of wireless sensor networks, the consequence of such an attack can be catastrophic to the operations of the entire network. A distributed denial of service attack may also culminate from other attacks such as Node Replication, Sybil as well as Node Implant attacks.

The distributed nature of such attacks in a wireless communication environment demand the need for a distributed attack detection mechanism in place. In this chapter, we propose an adversary model for a distributed denial of service attack in a wireless sensor network. This model defines distinct classes of adversarial nodes, based on their capabilities. The purpose of the
attack is to exhaust the limited energy resources of target nodes. We therefore model the energy consumption rates associated with the attack process, within the adversarial nodes, to illustrate the significance of detecting such attacks, before any catastrophic damage is done by such nodes, in the event of attack success. Based on the attack and the network model, a distributed attack detection scheme is proposed in Chapter 4, for detection of such attacks. This scheme relies on the collective detection process, achieved by a set of sensor nodes, referred to as the attack detector nodes, designated the task of attack traffic observation and reporting. The distributed nature of the attack, along with the topological placement of the target as well as the detector nodes, demands a distributed and collaborative mechanism, for attack detection. The complete attack detection process is based on the distinct differentiation between normal network traffic flow, and attack traffic flow, performed by the attack detector nodes.

We define a model for generating limits i.e. thresholds on the maximum numbers of traffic packets receivable by a given victim node, from a particular region of the network, in a given frame of time. These threshold values are subpatterns constituting a complete pattern of threshold values for a given victim node. Threshold values are generated based upon the topological placement in the network of both the attack detector nodes as well as the victim nodes. Distributed denial of service attack traffic is detected by the attack detector nodes, by comparisons of the stored threshold subpattern values with the observed traffic flow values.

In Section 3.1, we define the requirements for detecting distributed denial of service attacks in wireless sensor networks. In Section 3.2, the adversary model defining the various classes of adversarial nodes that may participate
in the attack is given. In Section 3.3, we model the three most common network topologies for wireless sensor networks, to facilitate pattern generation for attack detection purposes. We define threshold patterns for storage and analysis on detector nodes, in Section 3.4. We define a traffic flow observation table in Section 3.5, to facilitate the storage and update of observed traffic flow parameters by the detector nodes, for traffic packets flowing towards a given victim node. Each detector node holds a traffic flow observation table in its memory. In addition, the detector nodes also hold the threshold sub-pattern values i.e. traffic flow bounds, for each victim node. Finally we enlist the concluding remarks in Section 3.6.

3.1 Requirements for DDoS Attack Detection in Wireless Sensor Networks

Distributed denial of service attacks in wired networks are launched by the adversary class from multiple ends of an entire network, such as the Internet. This network comprises of both the malicious set of nodes, under the control of the adversary, as well as the set of victim nodes of the attack, performing their routine operations. The topology of a wired network limits the total number of entry points into the network of a victim node. Therefore, the malicious attack traffic, penetrating the victim node’s network, can be monitored and detected by the front-end nodes such as routers, firewalls and switches, of the network. Alternately, the attack can also be detected by detection systems on individual victim nodes.
In wireless sensor networks, the wireless nature of the communication media, accompanied with the limited energy resources of sensor nodes, differentiates distributed denial of service attack modeling and detection, in them. The adversary class monitors the flow of traffic in the network, and labels the more active nodes, in terms of transmitting and receiving data packets, as critical nodes, which need to be targeted as part of the distributed denial of service attack. We refer to all such critical nodes as target or victim nodes. The distributed denial of service attack is launched by the adversarial nodes towards these critical sensor nodes, from multiple ends of the network. The purpose of such attacks is to deplete the limited energy resources of the victim nodes. Furthermore, injected malicious nodes in the network steal the identities of the energy-depleted victim nodes, and participate, with malicious intent, in the network operations. The lack of a single entry point to the network makes the task of detecting these attacks more cumbersome.

The flow of traffic packets in wireless sensor networks follows a source-sink model (Culler et al., 2004), wherein sources (sensor nodes) generate sensory data packets, that need to be transmitted to a centralised base station, through a well-defined routing path. The topology of the wireless sensor network defines the network data delivery model. The topological designation of individual sensor nodes of the network, together with their placement, imply different expected traffic flow observations by each of the detector nodes. The detector nodes store a set of pre-generated traffic threshold values (subpatterns), defining the maximum numbers of packets a victim node may receive during a given period of time. We refer to these threshold values interchangeably as subpatterns or threshold subpatterns. These subpattern values are
generated at network initialisation time, based on specific topological placements of both the detector as well as the victim nodes, as defined in Section 3.4. A systematic concatenation of these subpattern values will generate an entire pattern of threshold values, defining the maximum numbers of traffic flow packets that may be destined for a given victim node from various regions of the network, in a given period of time. The attack detection process thus operates as a coordinated effort by a set of attack detector nodes, intending to reconstruct a complete pattern of observed traffic flow values, for comparison with the pre-generated traffic threshold pattern.

A single centralised entity can be designated the task of detecting anomalous traffic flow in the network. However, such an approach suffers from several drawbacks:

- the traffic flow may be outside the observation range of the detector node;

- a large set of threshold patterns will have to be stored and processed by the detector node, for each victim node of the network; and

- the lack of multiple interfaces on the detector node implies that the attack traffic flow may overwhelm the detector node itself, and thus disrupt the entire detection process.

The solutions proposed for distributed denial of service attack detection in high-performance networks are also not directly applicable to wireless sensor networks because:

- The lack of a single entry point to the wireless network demands the presence of multiple attack detector points to cover the entire network;
The adversary-class consists of adversarial nodes of varying capabilities, that need to be modeled individually;

The limited energy resources of sensor nodes cannot sustain any resource-demanding attack detection techniques on them; and

The distinct topologies of wireless sensor networks demand the need for definition of distinct patterns of normal network traffic, based on specific network topologies, to facilitate attack pattern detection.

In this chapter, we propose the following:

- An adversary (attack) model is proposed to define adversarial nodes based on the node capabilities;

- An adversary node energy usage model is defined to signify the potential impact of an attack;

- A network model is defined to classify wireless sensor networks into three distinct data delivery models;

- Distinct traffic threshold patterns, based on network topologies and data delivery models, are proposed; and

- A traffic observation table based on observed traffic features is defined in the detector node memory.

In Section 3.2, we classify the set of adversarial nodes in the network, based on their capabilities. In our proposed attack (adversary) model, we define attacker nodes based on their capabilities, and model the energy depletion rates within these attack nodes, to study the effect of the attack on their
resources. This model is used for analysing the success of the attack, in terms of the residual energy resources of the attack detector nodes, that will be used by them to operate as masquerade nodes in the network. We define the wireless sensor network model as a set of two types of nodes, namely, sensor nodes and the base station, where the base station has a few orders of magnitude more resources than a typical sensor node. Individual sensor nodes transmit their sensory readings to the base station via a well-defined routing path. We classify the sensor network into three classes of most common data delivery techniques, for threshold pattern generation purposes. It may be noted that the proposed attack detection scheme (Chapter 4) can operate on any underlying network topology. A detailed elaboration of these models is given in Section 3.3. We model the distributed denial of service attack in wireless sensor networks as a traffic threshold pattern in Section 3.4. The nature of such attacks demands the need for clear distinction of network traffic into attack and legitimate traffic, based on the network topologies defined in Section 3.3. We define a traffic flow observation table, to store the traffic threshold patterns, in Section 3.5. Finally, we enlist our concluding remarks in Section 3.6.

3.2 Adversary Model

The adversary-class is defined as a set of malicious entities, intending to inflict loss either directly, or through other entities, on the network. It is responsible for defining, and if need be, introduction of malicious nodes into the network, with the purpose of launching a distributed denial of service attack. The set
of malicious nodes intending to launch a distribute denial of service attack, can be classified into the following categories:

1. **Injected sensor nodes** again may consist of either sensor nodes with normal sensor capabilities, or more powerful sensor nodes, with the capabilities of say the base station.

2. **Compromised nodes** are defined as legitimate sensor nodes, whose operations are taken over by the adversary-class, for purposes of disrupting normal network operations.

3. **Laptop-class nodes** are defined as nodes with more communication resources, in terms of transmit and receive capabilities i.e. stronger antennas as compared to standard sensor nodes. In addition, laptop-class nodes have a battery supply sustaining the node for a longer lifetime as compared to normal sensor nodes.

In Figure 3.1, we illustrate a distributed denial of service attack model, in the presence of various types of nodes in the network. The legitimate nodes of the network include intermediary data aggregation (DA) nodes, cluster heads, non-cluster heads and the base station. The malicious nodes of the network include compromised nodes, malicious (injected) nodes and laptop-class adversarial nodes. The cluster heads, defined as sensor nodes, with added responsibilities, are labeled as target nodes in this particular example scenario. All other legitimate nodes are also vulnerable to a distributed denial of service attack, launched by the adversarial nodes of the network.

We address the problem of detecting attacks launched by the adversary-injected nodes in the network, and also scenarios wherein a set of sensor nodes
in the network may be compromised to launch flooding attacks against other legitimate nodes of the network.

The adversary-class launches the distributed flooding attack by instigating the malicious nodes in the network to generate a large set of attack packets from multiple ends of the network, towards the victim nodes. The success of the attack is achieved by the collusion feature of such an attack, where participation of multiple malicious nodes takes place. As a result, the per-node overhead incurred due to participation in the attack i.e. generation and transmission of large volumes of hoax packets by the adversarial nodes, is lowered significantly. The distributed flooding attack being launched from multiple-ends of the network, by multiple adversarial nodes incurs energy usage on the adversarial node, with total updated energy content of an adversary node $a_k$ at time $t_1$ given by: $E_{a_k}(t_1) = E_{a_k}(t_0) - E_{\text{trans}}(\frac{p}{k})$, for a $k$-adversary node network, with the summation of attack packets generated by all adversarial nodes = $p$. $E_{\text{trans}}(p)$ is defined as the energy usage for transmission of $p$
packets by an adversarial node. In case a single adversarial node is launching a flooding attack, the total amount of energy needed for transmission of \( p \) packets by this single adversarial node is given by: \( E_{\text{trans}}(p) \). The added saving of the total energy contents of the malicious nodes of the network when more than one adversarial node is present, facilitates the subsequent use of the adversarial nodes by the adversary class, for participation in further disruption activities in the network. These activities may include continuous transmission of flooding attack packets towards other nodes of the network, routing path disruptions, as well as message injection and tampering attacks.

The set of malicious nodes injected into the network by the adversary-class, need to communicate with each other for synchronous launch of a flooding attack. All such communication for adversary control operations takes place outside the communication band of standard sensor node communication channel, to avoid monitoring of adversarial activity as anomalies in communication channel usage, by the sensor network. The adversary class monitors the activity of the sensor network to handpick the most active nodes of the network. Therefore, sensor nodes participating in frequent reception and transmission of messages in the network are tagged as 'critical' nodes by the adversary class. We refer to these critical nodes as target (victim) nodes, and denote them as \( T = \{T_0, T_1, ..., T_{r-1}\} \). For instance, nodes closer to the base station, responsible for data forwarding to the base station from other nodes, will be more active in the reception and delivery of aggregated messages, and are therefore more likely to be labeled as target nodes by the adversary-class. The purpose of our proposed attack detection scheme (Chapter 4), is to detect distributed flooding traffic towards this set of \( r \) nodes of the network.
The tasks assigned to a sensor node, along with its topological placement in the network define the level of its criticality to the operations of the network. If the availability of a sensor node is essential for ensuring uninterrupted operations of the network, its identification and labeling as 'critical' by the adversary class can have catastrophic consequences to smooth network operation. The attackers upon identification of critical nodes, will launch a distributed flooding attack against them. The adversary class intends to exhaust the energy resources of the \( r \) identified target nodes belonging to the set \( T \), of the network, by simultaneous launch of flooding attack traffic towards the victim nodes. All other nodes \( N \notin T \), are less significant to the operations of the network, and therefore can be safely neglected for purposes of attack detection.

A distributed flooding attack can be considered successful, if the energy resources of the victim node(s) are exhausted due to the processing demands incurred on them for all operations related to the processing of this large scale influx of attack traffic packets. Upon complete exhaustion of energy in the victim nodes, the adversarial nodes may steal the identities of legitimate nodes in the network, to generate redundant or incorrect sensory data for delivery to the base station, degrade network performance by increasing packet drop rates and add to the delays associated with packet delivery to the base station. In addition, the adversary nodes may also launch further flooding attacks against other unaffected legitimate nodes of the network.

A second attack scenario involves the adversarial nodes launching the flooding attacks by masquerading as legitimate sensor nodes, and flooding the immediate network i.e. neighbor nodes, with large number of hoax requests. An ideal attack situation is when a set of colluding adversaries simultaneously
send a large number of hoax requests to the target node from multiple ends of the network. The attackers can thus act stealthily, where otherwise heavy traffic flow intended for a particular target node launched from a single end point of the network can be easily detected as a localised traffic anomaly by a single attack detector node operational in the specified region of the network. We denote the set of adversarial nodes in the network as: \( A = \{ A_0, A_1, ..., A_{k-1} \} \).

The reachability matrix of size \( a \times r \) defines the distances that need to be traversed by adversary-initiated messages for traversal towards the \( r \) victim nodes. The energy used for message traversal, \( E(a, r) \), is proportional to distance\((a, r)\), \( \forall \{a \in A \text{ and } r \in T\} \). The average energy utilisation rate, \( E_a \), of an adversary node \( a \in A \) is given by:

\[
E_a = \frac{1}{|A|} \sum_{i=1}^{|A|} \left( \frac{1}{v(i)} \sum_{j=1}^{|v(i)|} (E_{util}(i, v(i))) \right)
\]

(3.1)

where, \( v(i) \) is the set of victim nodes targeted by node \( a \), and \( E_{util}(i, v(i)) \) is the energy consumption for transmission of a malicious packet by a malicious node \( i \) towards a victim node \( v(i) \). The larger the victim node set for a given adversarial node \( a \), the higher the energy consumption rate will be. However, in the presence of a large number of adversarial nodes in the network, the per-node energy consumption rates associated with launching such an attack will be reduced. As a result, the lifetimes of the adversarial nodes will also be extended.

The average energy usage by a sensor node \( n \) intending to transmit a message to a destination node \( d \) in the network, is given by:
\[ E_n = \frac{1}{|N|} \sum_{i=1}^{\vert N \vert} E_{util}(n_i, d) \]  

(3.2)

where, both \( n_i \) and \( d \) are both sensor nodes, or either one of them is the base station.

The lifetime of the sensor nodes \( n \) of the network is given by: \( G(n) \), where: \( G(n) \propto \frac{1}{E_n} \), and the average lifetime of the adversary node in the network, \( G'(a) \) is given by: \( G'(a) \propto \frac{1}{E_a} \). Considering the frequent nature of packet generation and transmission by the adversarial nodes participating in the attack, for scenarios with fewer number of adversarial nodes, the individual lifetimes of the adversarial nodes, \( G'(a) \), can be considered to be \( << G(n) \). On the other hand, if a large set of adversarial nodes participate in the attack, we can expect: \( G'(a) >> G(n) \). It may thus be observed that the adversary-class participating in a distributed flooding attack will survive for longer, post attack success, as compared to scenarios, where centralised attacks are launched by the adversary class, through a single attacker node. As a result, the adversarial nodes can successfully masquerade as legitimate but victimised nodes, and operate unaffected in disrupting the operations of the network.

The distributed denial of service attack is thus more successful, if launched, in distributed fashion, by multiple adversarial nodes. It is therefore imperative to have a distributed attack detection mechanism in place to detect such attacks, by means of having multiple attack detector nodes operational in the network. In the next section, we model the topologies of a wireless sensor network, to facilitate generation of threshold patterns depicting bounds on the maximum receivable traffic by a victim node, in a given frame of time. These values are stored for subsequent analysis by the attack detector nodes.
3.3 Network model

The wireless sensor network model consists of a finite set of sensor nodes given by: $N = \{N_1, ..., N_n\}$, where $|N| = n$. The network also consists of a centralised base station in addition to the sensor nodes. The $n$ sensor nodes of the network consist of sensors with added capabilities and/or administrative and control tasks of the network (cluster heads and data aggregation points), as will be explained in the next paragraph. Victim nodes are defined as a set of nodes $T = \{T_0, ..., T_{r-1}\}$, where $T \subset N$, such that, each target node $r$ of set $T$ is a critical node of the network, and $|T| = r \ll n$. The adversary-class is defined as the set of malicious nodes in the network, and are denoted as: $A = \{A_0, A_1, ..., A_{k-1}\}$, where $|A| = k \leq n$.

Sensor nodes of a typical wireless sensor network operate with the purpose of monitoring and detecting events in their environments, for subsequent delivery of their respective observations and readings to a centralised base station. The data can either be delivered to the base station directly by the sensor nodes, or through a chain of defined intermediary nodes. The frequency of communication of messages by the nodes to the base station is referred to as the network taxonomy (Tilak et al., 2002). The data delivery model i.e. network topologies, defines the routing path for transfer of data from the sensors to the base station. We classify the data delivery model into three most common sensor network classes, namely, flat, cluster-based and data aggregation.

In the source-sink model of communication, the traffic packets originating from a source node can either be forwarded by the sensor nodes directly to the base station (flat topology), or through a set of intermediary nodes.
These intermediary nodes are referred to as aggregation nodes (Baek et al., 2004). The latter case can be further classified into cluster-based network topology and data aggregation-based network topology, respectively. The cluster-based topology relies on a two-hop approach for packet delivery by the sensor nodes to the base station, whereas the data aggregation-based topology uses multiple hops for packet delivery from the sensor nodes to the base station.

The three network topologies defined above are contemporary sensor network topologies. The distributed attack detection scheme proposed in Chapter 4 is an overlay, which operates independently, irrespective of the underlying network topology. The modeling of the network topologies is essential for definition of the target nodes of the network, which in turn will facilitate the generation of threshold patterns, that are required for attack detection (Section 3.4). The attack detection scheme can therefore operate beyond the three network topologies defined in this section.

The three network topologies, together with their traffic flow models are defined as follows:

1. Flat Topology: In a flat topology, each sensor node in the network directly communicates its sensor readings to the base station using a single-hop mechanism, without intermediate message transfer nodes to aid in the communication process. Every sensor node has equal priority designated for such networks. The traffic flow from the sensor nodes to the base station here can be expressed as $f = \{f_1\}$, depicting a single hop transmission to the base station. A flat-topology sensor network model is illustrated in Figure 3.2. It is assumed that for the flat topology
to operate successfully, all sensor nodes must have sufficient transmission ranges to facilitate their communication with the centralised base station.

Figure 3.2: Flat Topology

2. Cluster-based Topology: In a cluster-based network topology (Figure 3.3), a set of sensor nodes with added capabilities are defined as cluster-heads. These cluster-head nodes act as control and administrative centres for a set of pre-defined clusters of sensor nodes in the network. Cluster heads are responsible for the administration of their respective clusters, data aggregation from sensor nodes of their clusters, and data forwarding to the base station. In addition, cluster heads are also responsible for monitoring the status of sensor nodes in their clusters, and reporting of faults and losses to the base station. Cluster-based networks generally follow a two-hop traffic flow path to reach the base station. This flow can be expressed as: $f = \{ f_{f,ch(f)}, f_{ch(f),bs} \}$, where $f_{f,ch(f)}$ is the flow from node $f$ to its cluster head $ch(f)$, and $f_{ch(f),bs}$ is the flow from $ch(f)$ to the base station. Certain special-case cluster-based topologies rely on multiple hops for data transfer between the
cluster heads, before being forwarded to the base station. We classify such topologies as data aggregation network topologies.

![Cluster-based Topology](image)

Figure 3.3: Cluster-based Topology

3. Data Aggregation Topology: In a data aggregation topology (Figure 3.4), sensory readings from individual sensor nodes progress through the network from the source node towards the base station, through a well-defined tree of interconnected intermediary nodes. The data along the path is aggregated at specific nodes in the network called aggregation points, defined as nodes with the numbers of incoming edges to the nodes exceeding their total outgoing edges (usually equal to unity). The purpose of aggregating intermediary data is to reduce the total traffic flow in the network, and to minimize the energy consumption associated with frequent and large-scale data transfer operations to the base station by individual sensor nodes of the network. A typical data aggregation topology consists of interconnected trees defining the flow of network traffic from individual source sensor nodes to the base station. The traffic flow from the sensor nodes of a data aggregation tree through
aggregation nodes can be expressed as \( f = \{ f_1, f_2, \ldots, f_{L(f)} \} \), where \( L(f) \) is the length of path from node \( f \) to the base station.

![Diagram of Data Aggregation Topology](image)

**Figure 3.4: Data Aggregation Topology**

The network model defined in this section is crucial for the generation of traffic threshold patterns for attack detection purposes. The threshold pattern generation process relies on the underlying topology of the network for generation of subpattern values for storage and subsequent comparison by the attack detector nodes.

### 3.4 Threshold Pattern Modeling

In the previous section, we have classified wireless sensor networks into three most network topologies based on the data delivery models. In this section, we propose the generation of threshold subpattern values for storage and comparison within the attack detector nodes, based on these defined network topologies.

The analytical model of a sensor network undergoing a DDoS attack consists of two types of network traffic, namely, normal and attack. The flow
of traffic in a typical sensor network is directed from the sensors to the base station. During normal operations mode, a sensor node may receive traffic from several sources, such as from nodes within its immediate vicinity. The volume of traffic, and in essence, the numbers of traffic flows is higher, if the sensor node is a cluster-head or a data aggregation node. The traffic constituting a distributed denial of service attack can also be categorised as a flow, albeit with a different label. We assume that each adversarial node generates a single flow of traffic towards a victim node \( r \). In the presence of attack traffic, the total traffic received by a target node \( r \) in a given time epoch, and that needs to be monitored by the attack detection scheme, is given by:

\[
\lambda_r = \sum_{i=1}^{f} \lambda^i_{r,i} + \sum_{j=1}^{k} \lambda^j_{r,j}
\]  

(3.3)

where \( \lambda^i_{r,i} \) is the normal traffic rate belonging to traffic flow from node \( i \), and \( \lambda^j_{r,j} \) is the attack traffic rate originating from an attacker node \( j \) belonging to the attacker set \( A \). Each node in the network is considered to bear a single queue, with average time for packet processing and transmission at node \( i \) being \( s_i \). The intensity of the arriving traffic at node \( r \) is thus given by:

\[
\rho_r = s_i(\sum_{i=1}^{f} I^i_{r,i} + \sum_{j=1}^{k} I^j_{r,j})
\]  

(3.4)

\( I^i_{r,i} \) is defined as normal traffic intensity, whereas \( I^j_{r,j} \) is defined as the attack traffic intensity for all attack nodes \( k \in A \). We consider the case of attack detection by means of studying the overall traffic intensity towards a set of target nodes in the network. The traffic arrival intensity at node \( r \) thus is a function of the individual arrival intensities of both the normal sensory traffic, as well as the attack traffic.
A distributed denial of service attack flow is launched by several attacker nodes from multiple ends of the network. The attack packets may arrive at the target node(s) from different regions of the network, and therefore a collaborative effort is needed to detect distributed anomalous traffic flow towards the target node set. We define a set of sensor nodes called attack detector nodes, as nodes which observe traffic flow of the network towards the target node set $T$. These nodes are notated as: $G = \{g_0, g_1, ..., g_{d-1}\}$, where $|G| = d$. The broadcast nature of traffic in sensor networks facilitates the promiscuous monitoring of traffic flows in the network towards the target nodes. Each of these detector nodes is responsible for storage of a single threshold (subpattern) value, for each of the $r$ target nodes, depicting an estimate on the number of requests receivable by a target node during a fixed interval of time $\Delta$. These threshold values are defined as the maximum numbers of packets a node $r$ is willing to accept from a particular network region, during a constant time interval $\Delta$, from the region of operation of the observer (detector) node. One of the factors for generating these threshold values is the the topological designation of a target node in the network. We define distinct attack patterns based on this topological placement of the sensor nodes. Considering that different threshold (subpattern) values will be stored in different attack detector nodes, the complete threshold pattern vector is a unique pattern defining a set of bounds on the receivable traffic by a node $r$ during a given time interval, from all regions of the network. For a constant network taxonomy, the total traffic that a particular node $r$ in the network can expect in a given time interval is denoted as $P_r$. This expected traffic inflow value depends on the network taxonomy, node $r$’s initial energy content, its expected lifetime, and the average energy resource usage by node
$r$ for processing of each received packet. These values are generated beforehand at network initialisation time, and are constantly updated based on the current energy contents of the target nodes.

The pattern generation criteria varies for the three network topologies defined in Section 3.3. Each network topology has a different set of selected potential target nodes $T$ in the network. Each target node has a different set of traffic flow patterns towards them, that need to be observed. In a flat topology, all sensor nodes in the network are at the same level of criticality. The loss of any of the $n$ nodes in this topology is likely to have an equal impact on the operations of the network. Subpattern (threshold) values for a flat topology are generated based on Equation 3.5. The threshold subpattern values for target node $r$, stored in the detector node $d$, is denoted as $\text{th}_{r}^{d}$.

Sensor nodes are deployed at network initialisation time. The base station has a record of the total number of nodes in the network, as well as an estimate on the distances to each of these nodes. The above parameters facilitate computation of the density of node deployment. For instance, a network spanning a large geographic area with fewer numbers of nodes, will have a low node deployment density, and a network covering a smaller geographic area, with large numbers of deployed nodes, will have a higher node deployment density. The density of deployment of nodes in a flat topology defines the extent of loss that may be incurred on the network due to the loss of a single target node. For denser networks, the loss of a few nodes will be less significant as compared to a network with low node deployment densities. Therefore, the observable threshold value, $\text{th}_{r}^{d}$, is high for denser networks, implying that a larger set of target nodes can be lost before an alarm can be raised.
\[ th^r_d = \left[ P_r + nw(density) + \frac{1.0}{d_{G(d)(r)}} \right] \]  

(3.5)

where,

\( nw(density) = \) Normalised node deployment density of the network.

\( d_{G(d)(r)} = \) Normalised Euclidean distance from detector node \( d \) to the target node \( r \).

\( P_r = \) Normalised number of expected packets by node \( r \) in a fixed interval of time, \( \Delta \).

The Euclidean distance of a target node from a particular detector node is another factor used in the computation of the threshold pattern value. Target nodes outside the observation range of a detector node, \( d \), need to be monitored by other closely located detector nodes in the network. A lower threshold value implies fewer numbers of traffic packets are expected from this particular region, towards the target node.

In a cluster-based network topology, the cluster heads play a crucial role in the operations of the network, and therefore, need constant monitoring of traffic flow towards them. We therefore consider the cluster heads to be critical nodes in this topology. The threshold subpatterns for this network topology are generated from Equation 3.6.

\[ th^r_d = \left[ P_r + num_{ch} + \frac{1.0}{d_{G(d)(r)}} \right] \]  

(3.6)

where,

\( num_{ch} = \) Normalised number of clusters in the network.

\( d_{G(d)(r)} = \) Normalised Euclidean distance from detector node \( d \) to the target node \( r \).
The value of $d_{G(d)(r)}$ is the normalised distance between the cluster heads and the detector nodes. This value defines the expected traffic flow intensity towards the cluster heads from different regions of the network. Cluster heads distantly located from the base station are generally at the end of a tree routing hierarchy, and thus accumulate fewer numbers of traffic packets from leaf-end sensor nodes, in a given time interval. The values of $P_r$ are lower for such nodes. On the contrary, cluster heads closer to the base station are responsible for aggregation of packets, in addition to their cluster head operations, and therefore expect higher traffic inflows, due to the influx of large cumulative traffic payload. The values set for $P_r$ are higher for such nodes. The normalised Euclidean distances between detector nodes closer to cluster nodes yield higher threshold values, depicting more numbers of expected requests towards these cluster nodes, whereas detector nodes farther away from a cluster node are considered to be outside their respective regions of monitoring, and therefore lead to reduced threshold values. The node deployment density in cluster-based networks defines the total numbers of operating clusters in the network. Therefore, for higher node deployment densities, higher values of $th^*_d$ are generated, indicating lesser significance given to each cluster node.

In a data aggregation topology, the data aggregation nodes in the network are significant in the aggregation and forwarding of sensory data up the tree hierarchy. The loss of these nodes may lead to the inactivity of a complete arm of operation (sensor region) of the network. Data aggregation nodes are considered critical target nodes in this topology. The pattern generation equation for a data aggregation topology is:
where,

\[ d_{G(d)}(r) = \text{Euclidean distance from detector node } d \text{ to the target node } r. \]

The density of node deployment plays a less significant role in data aggregation networks, as the tree paths for routing of sensory data are fixed at network initialisation time, and remain unaltered. The number of hops separating a data aggregation node from the base station define its level of significance. Aggregation nodes closer to the base station will expect more inflows of network traffic towards them, and therefore will have higher \( P_r \) values associated, thus leading to higher threshold values. On the other hand, aggregation nodes closer to the leaf-end sensor nodes will expect lesser traffic inflow from sensor nodes lower in the hierarchies, and therefore, will have smaller associated \( P_r \) values, thus leading to smaller threshold subpattern values. Detector nodes in proximity to the data aggregator nodes are expected to observe higher traffic flow towards them, whereas detector nodes farther away from the aggregation nodes set lower \( th_d^r \) values indicating fewer traffic flow rates towards the target node.

All communication packets in the network are assumed to have a node identification tag appended to them for identifying both the source as well as the intended destination of the traffic packets. Node identification can be generated using unique knowledge possessed by a sensor node. Such knowledge can be the relative geographic location of the sensor node, which can be preset into the sensor memory at network initialisation time. The ID for node \( n \) at location \( <l_x(n), l_y(n)> \) is given by: \( \Lambda : N \rightarrow N(l_x, l_y) \), where function
A uses the geographic coordinates of a node \( n \), to derive its unique location coordinate identifiers.

In Table 3.1, we have illustrated the sub-pattern (threshold) values that need to be stored in each of the \( d \) detector nodes of the network, along with the location coordinates of the target nodes \( l_x(r), l_y(r) \), to facilitate feature comparison associated with real-time traffic flow, with stored threshold sub-pattern values. The complete pattern vector, if to be analysed by a centralised entity, for a given target node \( t_1 \), in the presence of \( d \) detector nodes, is given by: \( < l_x(t_1), l_y(t_1), th_{01}, th_{11}, ..., th_{1d} > \).

Threshold subpatterns given in Table 3.1 must not be exceeded by monitored network traffic during a given time frame, \( \Delta \). Each attack detector node is responsible for storing a single subpattern value for each target node. Although the individual observation of a single detector node will not depict an entire flooding attack scenario, the coordinated reconstruction of the complete pattern of observed traffic readings, by all detector nodes, facilitates achieving the same.

<table>
<thead>
<tr>
<th>(Detector Node, Node ID)</th>
<th>( t_1 )</th>
<th>( t_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, ID(1))</td>
<td>( th_1^1, l_x(t_1), l_y(t_1) )</td>
<td>( th_1^2, l_x(t_2), l_y(t_2) )</td>
</tr>
<tr>
<td>(2, ID(2))</td>
<td>( th_2^1, l_x(t_1), l_y(t_1) )</td>
<td>( th_2^2, l_x(t_2), l_y(t_2) )</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>(d, ID(d))</td>
<td>( th_d^1, l_x(t_1), l_y(t_1) )</td>
<td>( th_d^2, l_x(t_2), l_y(t_2) )</td>
</tr>
</tbody>
</table>

Table 3.1: Threshold subpatterns for a set of two example target nodes, to be stored one each within the \( d \) detector nodes.
3.5 Traffic Flow Observation Table

After the expected initial threshold values for a set of target nodes are generated and stored in the detector nodes, the attack detection scheme, proposed in Chapter 4, does comparisons of statistical features extracted from observed traffic flow in the network. These features define the intensity of traffic flow in the network towards a set of $r$ target nodes, for classification of flooding attacks by the attack detection scheme. The features to be extracted from the traffic constitute the pattern vectors that need to be compared during the pattern matching process of the detection scheme. These traffic features are given by:

- Percentage of packets with destination address = $d$, where $d \in T$.

- Percentage of packets with source address = $\{s \mid \forall_r, \text{Euclidean}(s, r) > thr_{euc}\}$, where $thr_{euc}$ is the threshold on maximum permissible distance between the detector and the target nodes.

- Percentage of packets with source address = $\{s \mid s \notin \text{cluster}_d, \text{where } d \in T, s \in N\}$.

The attack detector nodes need to observe and analyse the set of packets that are intended for any of the nodes in the target node set. Therefore, the destination address of each packet is an important feature, required for attack detection. The second feature defines the significance of including a particular packet in the traffic analysis done by the detector node. A packet intended for a target node, at a higher than threshold Euclidean distance, is analysed by other detector nodes, within the target node’s vicinity. Similarly, packets originating from outside the cluster of operation of a detector node,
in a cluster-based network topology, need to be analysed by other detector nodes.

**Definition 3.1.** \( \forall \) patterns \( p_r \), \( \text{length}(p_r) = 2r \)

If we assume a centralised approach towards attack detection i.e. without the presence of localised decision making in the network, the total number of pattern vectors expected by a base station for classification purposes, at the end of a time epoch \( \Delta \) is equal to: \( n \). The length of each pattern vector, as can be seen from Figure 3.5, is equal to \( 2r \).

![Figure 3.5: Pattern vectors reconstituted for comparison with predefined threshold values.](image)

For each of the \( r \) target nodes, a pattern vector, \( p_r \), will be reconstituted at the base station based on the receipt of individual subpatterns from each of the \( n \) detector nodes. The pattern vector for target node \( r \) is given by: \( p_r \)
\[ \{p^r_1, pe^r_1, p^r_2, pe^r_2, \ldots, p^r_n, pe^r_n\} \]

where \( p^r_n \) is the percentage of packets destined for target node \( r \), observed by detector node \( n \), and \( pe^r_n \) is the percentage of packets observed by node \( n \) as to possessing a source address outside the Euclidean threshold defined by \( thr_{euc} \), as satisfying the second rule for feature extraction defined above.

For cluster-based wireless sensor networks, the pattern vector for a target node \( r \) is given by: 
\[ p_r = \{p^r_1, pc^r_1, p^r_2, pc^r_2, \ldots, p^r_n, pc^r_n\} \]

where \( p^r_n \) is the same as the previous scenario. However, for cluster-based networks, we define the sub-pattern \( pc^r_n \) as a value indicating the percentage of packets observed by detector node \( n \) as being directed to a target \( r \) from outside its cluster of operation, \( cluster_r \).

We have illustrated above the techniques for generation of pattern vectors from observed real-time network traffic flow. These pattern vectors are compared with the threshold subpattern values, generated and stored in each of the attack detector nodes. The attack subpatterns vary based on several parameters including the proximity of the target nodes to the detector nodes. Therefore, the threshold values generated are different for storage on each of the detector nodes, and cannot be modeled simplistically as a cumulative sum of all subpatterns i.e. traffic flow values, towards a target node during a given time epoch \( \Delta \).

The attack detection scheme proposed in Chapter 4 addresses the need for localised reconstruction of patterns from individually observed traffic sub-pattern values by each of the detector nodes.
3.6 Conclusions

In this chapter, we modeled distributed denial of service attacks in wireless sensor networks. We illustrated the need for having multiple sensor nodes, with added responsibilities, to detect such attacks, when launched from multiple ends of the network, by adversarial nodes. We defined distinct classes of adversarial nodes that may launch such attacks. We analyzed the energy resource usage associated with the launch of these attacks, by the adversary class, and concluded that if the attacks are launched in distributed manner, from multiple ends of the network, they will prove to be more successful, as compared to scenarios where the attack is launched from a single front, by a single adversarial node.

A network model was defined to classify wireless sensor networks into topologies, based on the source-sink data delivery model. Three distinct classes of wireless sensor networks were defined, and the traffic flow, inclusive of both attack and normal traffic, was defined for each topology, separately. We also classified a set of legitimate sensor nodes as target (victim) nodes in each topology, based on the significance of the nodes to the network operations. We modeled the attack based on the expectation that an attack launched by the adversary class against these target nodes will prove to be more disruptive.

We proposed a model for expected traffic flow towards the victim node set, based on several criteria, namely, the node deployment densities, proximity of the target nodes to the base station, and the proximity of the target nodes to the detector nodes. These parameters facilitate generation of a sequence of threshold subpattern values, that depict bounds on the maximum
traffic flow, permissible towards a given target node, during a fixed interval of time. The defined threshold values are stored in the d attack detector nodes of the network. All detector nodes also maintain a traffic observation table, defined in Section 3.5, in their local memory. This table is updated with the observed traffic flow towards the victim node set in each frame of time, ∆. Subsequently, the updated values from the traffic observation table are compared with the previously generated traffic threshold subpatterns, to decide on whether the observed traffic can be labeled as anomalous in nature, or not.

A single traffic observation value will not generate a conclusive decision on an attack. However, a complete reconstruction of the traffic observation pattern, constituting of these subpattern values, will facilitate in the decision making process.

In Chapter 4, we define the complete attack detection scheme, which uses the threshold subpattern values, and the traffic observation table defined in this chapter, for observed traffic classification. The scheme proposed in Chapter 4 is for the first two classes of adversarial nodes, namely, injected nodes in the network. In Chapter 5, we perform a quantitative analysis of results acquired from simulation experiments, to study various aspects of the proposed scheme. In Chapter 6, we propose a fault-tolerant approach towards attack detection, to operate with high success rate, in the presence of compromised nodes in the network (third class of adversarial nodes). We also perform a quantitative simulation analysis of the proposed scheme.
Chapter 4

Distributed Attack Detection Scheme

In wireless sensor networks, a distributed denial of service attack can be launched by one of three adversary types, namely, injected nodes, laptop-class nodes, and compromised nodes, as described in Chapter 3. These malicious nodes participating in the attack, exploit the lack of a single entry point in a wireless network, and generate large volumes of malicious traffic, from multiple-ends of the network, towards a set of victim nodes. The purpose of such attacks is to overwhelm the limited energy resources of target sensor nodes, thereupon replacing legitimate nodes with malicious nodes, with the intent of disrupting network operations.

For distributed monitoring and attack detection, there exists a need for an efficient and accurate mechanism in place, to successfully recognise patterns of anomalous network traffic flow. A centralised approach towards attack detection in such scenarios will incur significant delays, associated with constant monitoring of attack traffic by a single designated node, and will thus
reduce the effectiveness of the outcomes of the detection process, on post-
attack mitigation techniques, such as, resource reallocation and topological
reorganisation of the network by the base station.

In this chapter, we propose a distributed attack detection scheme, based
on distributed pattern recognition, applicable for scenarios wherein the adver-
sary class consists of malicious nodes injected in the network for large-scale
traffic generation, directed from multiple-ends, towards a target node set.
Distinct threshold patterns of normal network traffic flow, are predefined for
the three common sensor network topologies, using the pattern generation
criterion defined in Chapter 3. The attack traffic flow towards the set of
victim nodes of the network is monitored by a set of selected attack detec-
tor sensor nodes. The detector nodes also collaborate and exchange their
individual findings with peer detector nodes at regular time intervals. The
decision making process is accomplished by a subset of these attack detector
nodes, selected by the base station at network initialisation time, to perform
additional tasks appertaining to attack decision-making.

Unlike centralised attack detection, wherein a single node will be responsi-
ble for all tasks associated with the detection of an attack in a wireless sensor
network environment, thus establishing a single point of failure, a distributed
approach will prove to be more reliable. In addition, a distributed scheme will
balance the tasks associated with the attack detection process over the entire
set of detector nodes, rather than overwhelming a single centralised node.
As a result, the per-node energy utilisation incurred by the attack detection
process will reduce.
4.1 Introduction

The varying topologies of a wireless sensor network, impose varying demands on the sensor nodes. Some nodes, such as cluster heads, may have more tasks assigned to them, as compared to others. In Chapter 3, we modeled the traffic flow in the network by defining the maximum number of receivable packets by a given sensor node, based on the topological designation of the node, as well as the proximity of the node to a detector node. These set of values constitute a pattern, defining a holistic view of acceptable traffic packet rate by a victim node, per unit of time. Since these individual traffic rates define bounds or thresholds on the total number of incoming packets to a given node, we also refer to these subpattern values as threshold subpatterns.

The threshold subpatterns vary depending on the topological placement of both the attack detector nodes, as well as the target nodes in the network. These subpattern values thus vary depending on the tasks assigned to a victim node, as well as the proximity of the victim nodes to the detector nodes. Therefore, there exists a need for having a distributed and collaborative mechanism in place to perform attack pattern recognition. This process of recognising patterns in network traffic flow, must include successful reconstruction of patterns depicting observed network traffic (observed traffic subpatterns), from various ends of the network, by the detector nodes, for comparison with a set of predefined threshold subpatterns. This comparison is essential for confirmation of an attack in progress against target sensory resources.

Our scheme performs in-network, distributed pattern recognition for distributed denial of service attack detection. The scheme is multi-tiered and
distributed in nature (Fig. 4.1), and consists of three layers of operation. The first layer of the scheme consists of a set of sensor nodes imposed with an additional task on them, of attack traffic monitoring and subsequent coordination with peer detector nodes. These nodes follow the emergent property of a network or system (Gligor, 2004), wherein the nodes, on an individual basis, will not be able to achieve the objectives of attack detection, but rather would rely on the collaboration amongst all such nodes, for pattern reconstruction and attack confirmation. Since these nodes follow the characteristics of the distributed pattern recognition algorithm, namely, the Graph Neuron (GN), we refer to these nodes as Graph Neuron or GN nodes. In addition, we will also interchangeably refer to them as attack detector nodes.

Figure 4.1: Multi-tiered Overlay for Distributed Attack Detection; Layer 1: GN nodes, Layer 2: mGN nodes and Layer 3: Base Station.

Normal network traffic flow towards the victim node set, is upperbounded by the threshold subpattern values, defined based on network topological
criteria, at time of network initialisation. These threshold subpatterns are
stored within the attack detector nodes. In Section 4.2, we elaborate on sev-
eral network-level factors that affect the values of the subpatterns generated.

In our proposed distributed attack detection scheme, each GN node ob-
serves network traffic flow, and generates a traffic observation subpattern for
each target node, regularly at the end of each fixed time interval or epoch,
of length $\Delta_{opt}$. The observed traffic flow in the network is compared by the
GN nodes with predefined thresholds of maximum number of traffic pack-
quets, receivable by each of the victim nodes of the network during the same
length time epoch. The GN nodes further collaborate with peer GN nodes,
to reconstruct an entire pattern of traffic flow observation, to confirm, at a
holistic level, an attack in progress. The collaboration and message exchange
between the GN nodes facilitates the verification of the traffic observations
of individual attack detector nodes. In addition, the threshold pattern values
within each GN node, are regularly updated based on the rate of decline of
energy resources within the victim nodes, to reduce the acceptable packet
limit of the node, so as to be sustainable for its expected lifetime.

The second layer of the scheme consists of a set of localised decision-
making nodes called master Graph Neuron (mGN) nodes, where the mGN
nodes belong to the set of GN nodes. Individual GN nodes upon coordi-
nation and communication with neighbouring GN nodes, communicate their
observed outcomes to their designated mGN nodes, at the end of each epoch
of time. These mGN nodes are responsible for taking a localised decision on
whether an attack is in progress or not, against any of the victim nodes of
the network. All verdicts issued by the mGN nodes are communicated to the
base station (layer 3), for further action.
The proposed detection scheme can operate as an overlay on any underlying network topology, provided that the subpatterns depicting thresholds on normal network traffic flow are defined for the given topology at the time of network initialisation.

4.1.1 Preliminaries

Wireless sensor networks can consist of both static and mobile sensor nodes. However, most sensor networks consist of sensor nodes, which remain static post-deployment. Therefore, node locations remain unchanged at network initialisation time by the base station. Individual sensor nodes also have a unique identification tag assigned to them at network initialisation time. This tag is used for marking the addresses of sources and destinations of all generated data packets. The generation of node identification tags can be accomplished by using a unique characteristic or possessed knowledge of a sensor node. For our scheme, we define the location identifier of a sensor node as a function of its relative geographic location in the network. Sensor nodes are pre-configured with geographic coordinates by the base station at initialisation time.

Detector (GN) nodes of the scheme are pre-configured with the knowledge of their neighboring sensor nodes, to facilitate distributed pattern recognition for attack detection. For this purpose, the detector nodes are preset with location details of their neighbor nodes within the network. The GN nodes are also loosely time synchronised to ensure timely and accurate completion of the communication phase of the GN application.
4.1.2 Contributions

The contributions of this chapter are as follows:

- A distributed, pattern recognition scheme for DDoS attack detection in wireless sensor networks is defined:
  - Pattern learning performed by the GN nodes at network initialisation time based on the topological placement of the target nodes in the network, using Equations 3.5, 3.6 and 3.7.
  - Pattern update is performed based upon traffic inflow i.e. sensor energy consumption rates within the target nodes.

- A distributed algorithm is proposed for selection of the mGN decision-making nodes at network initialisation time.

- An optimal time epoch length, $\Delta_{opt}$, formulation is proposed to achieve tradeoff between frequent attack detection and rapid energy resource exhaustion in the detector and the decision-making nodes.

- Formulation of an analytical model to analyse the overhead incurred by the scheme in terms of delay and energy resource usage.

In Section 4.2, we define the attack detection scheme. The scheme does in-network pattern recognition, to achieve the objectives of attack detection. All phases of operation of the scheme need to be executed regularly, during each epoch of time. We define a tradeoff formulation in Section 4.3, for computation of the optimal time epoch lengths for the detection scheme. In Section 4.4, we define a distributed algorithm for selection of mGN nodes, based on two criteria, namely, node deployment density and node reachability.
(in terms of communication range). The length of the time epoch will affect the accuracy of the detection scheme. A qualitative analysis of the efficiency of the attack detection scheme is given in Section 4.5. Finally, we enlist the concluding remarks in Section 4.6.

4.2 Attack Detection Scheme

As part of the attack process, malicious nodes generate a large set of hoax packets, for transfer towards a set of selected target nodes i.e. nodes at critical locations of the network. These attacker nodes may belong to the existing network, in essence, implying compromised but legitimate sensor nodes. In addition, the nodes may also be injected into the network by the adversary class, for purposes of participating in the attack traffic generation process. Upon successful attack completion, these injected nodes can replace the legitimate nodes of the network, and generate false (misleading) sensory data for transfer and delivery to the base station. An attack launched from a single entry point of the network is observable by a single detector node. On the contrary, a distributed denial of service attack requires a coordinated effort by a set of detector nodes, present at various locations of the wireless sensor network, so as to accurately detect such attacks. Our attack detection scheme proposed in this section performs detection of such attacks, when they are launched by both injected sensor nodes as well as laptop-class nodes. A scheme for detecting such attacks in the presence of compromised sensor nodes is defined in Chapter 6. The notations for our scheme are enlisted in Table 4.1.
The attack detector nodes promiscuously monitor traffic packets generated and/or transiting through their respective local neighbourhoods. These nodes are also programmed to coordinate and exchange traffic observation messages with neighbouring (peer) GN nodes, for pattern reconstruction and traffic observation verification purposes. For instance, the GN node $G N_i$ will exchange its traffic observation subpatterns with the GN nodes $G N_{i-1}$ and $G N_{i+1}$, respectively (Figure 4.1). The message exchange process is performed by each GN node, once during each interval of time, of fixed length equal to $\Delta_{opt}$. The purpose of exchange of the traffic subpattern values is: a) verification of the observed readings with peer GN nodes, and b) reconstruction of partial patterns of observed traffic, from peer traffic observation readings. Upon reconstruction, the complete observed traffic pattern for a given target node $r$ is defined as the concatenation of all observed subpattern values, $p^r_1, p^r_2, ..., p^r_n$ (Fig. 3.5), observed by the GN nodes. For instance, subpattern
$p^r_n$ defines the total observed traffic packets by GN node $n$, destined for target node $r$, within the current time epoch.

The threshold subpatterns stored within the threshold table (Table 3.1) of each GN node depict the maximum packets receivable by a victim node $r$ from the region of operation of GN node $n$. Thus, each GN node holds exactly $r$ threshold subpattern values at any given time, one for each of the $r$ target nodes. The traffic observation subpattern values define the total number of traffic packets observed by a GN node, destined for a given target node, during the current time epoch. These observed subpattern values are compared with corresponding subpattern values in the threshold table, for localised confirmation of anomalous traffic intensities, by each GN node.

With the progression of the actual network time, the target nodes will have receding energy content values, owing to sensory operations associated with processing both normal as well as malicious packets. Considering this recession in the total energy content values of the target nodes, the total number of message packets receivable by these nodes in the same time interval length $\Delta_{opt}$, must decrease, to ascertain that the node survives its expected lifetime. As a result, the threshold subpattern values for each of the $r$ target nodes, stored in the threshold tables of the GN nodes, need to be updated on a regular basis, to reflect these reducing numbers of requests receivable by the target nodes.

Upon successful exchange and reconstruction of the subpatterns for each of the $r$ target nodes of the network, exactly half of the total number of GN nodes of the GN array, communicate with their respective designated master nodes, called mGN nodes. The purpose of having only half of all the GN nodes communicate with their mGN nodes is to: a) avoid duplication in the
messages received by the mGN nodes, and b) reduce the overall communication overhead associated with more number of message transmissions, as part of the detection scheme.

The mGN nodes are a subset of the GN node set, selected by the base station, at network initialisation time, based on the \textit{mSelect} algorithm, proposed in Section 4.4. The purpose of having mGN nodes is for collection of individual traffic observation messages from the GN nodes, and generation of a verdict signal confirming an attack in progress against any or all of the \( r \) target nodes. The proximity of the mGN nodes to the GN nodes reduces the overall communication overhead associated with the frequent transfer of observation messages from the GN nodes, directly to the base station. The number of mGN nodes \( m \) is much less than the total number of GN nodes in the network, \( n \). Having a large number of mGN nodes in the network will increase the overhead on each individual node. This is because of the additional tasks of data collection and forwarding, performed by the mGN nodes. The existence of few mGN nodes for the detection scheme will increase the overhead on the smaller set of mGN nodes. However, the overall number of nodes that need to perform additional mGN-related tasks is consequently reduced. In Section 4.4, we propose an algorithm, for generation of a minimal set of mGN nodes based on the criterion of communication connectivity between the GN and the mGN nodes. The set of mGN nodes is a subset of GN nodes, with guaranteed reachability between each mGN node and the set of GN nodes in its respective jurisdiction.

During each epoch of time, the mGN nodes upon receiving reconstructed subpatterns from their designated GN nodes, generate a message for delivery
to the base station, depicting either an attack against a target node, or a normalcy signal indicating smooth network traffic flow.

The base station is responsible for taking a final decision as to whether an attack is in progress against a target node. The steps associated with our proposed distributed flooding attack detection scheme can be subdivided into five phases of operation, namely:

1. Initialisation
2. Observation
3. Communication
4. Verdict
5. Pattern Update

Apart from Phase 1 i.e. Initialisation phase, all other phases of the proposed scheme need to be executed within each interval of time, of fixed duration: $\Delta_{opt}$.

In Fig. 4.2, we illustrate the attack detection process by means of a flowchart. During the Initialisation phase of the scheme, sensor nodes are selected by the base station to operate as GN nodes to participate in the detection process. A subset of these GN nodes is then selected, based on the $m$Select algorithm (Section 4.4), to serve as the mGN decision-making nodes of the scheme. In addition, the optimal time epoch lengths, based on the formulation given in Section 4.4, are computed and pre-configured within the GN and mGN nodes, respectively. The equations defined in Section 4.4 define a tradeoff between frequent attack detection, and the lifetimes of the attack
detector nodes. These time epoch lengths facilitate time synchronisation between the GN and the mGN nodes, for correct and smooth functioning of the detection process.

The GN nodes monitor and observe traffic flow towards the r victim nodes, during the Observation phase of the algorithm. A traffic flow observation table, Figure 3.5, within each of the GN nodes is updated during this phase. Subsequently, the GN nodes communicate with each other to exchange their observations, during the Communication phase of the detection scheme. The mGN nodes generate their attack or normalcy signals delineating an attack in progress, or normal traffic flow observations towards any or all of the victim nodes, during a given time epoch $\Delta_i$, in the Verdict phase, for delivery to the base station. The GN nodes update their locally stored threshold subpattern values appertaining to each of the target nodes, during the Pattern Update phase of the algorithm. All nodes return to their Observation phase upon completion of the pattern update process.

In Algorithm 4.1, we illustrate the following five phases of execution of the attack detection scheme:

4.2.1 Phase 1: Initialisation

The initialisation phase of the detection scheme is completed at the end of the actual initialisation of the sensor network, performed by the base station. During this phase, node identification tags and topologies are established. The scheme initialisation consists of the following two sub-phases:
Figure 4.2: Phases of the attack detection scheme. Phase 2-5 are executed in each time epoch $\Delta_i$. 

- **Phase 1: Initialisation**
  1.1: mGN/GN node selection
  1.2: Time epoch calculation
  1.3: Pattern learning

- **Phase 2: Observation**
  2.1: Monitor traffic flow
  2.2: Update traffic flow tables

- **Phase 3: Communication**
  3.1: Inter-GN message exchange
  3.2: GN-mGN message exchange

- **Phase 4: Verdict**
  4.1: mGN-Base station communication
  4.2: Decision making

- **Phase 5: Pattern Update**
  5.1: Update threshold subpatterns
Detector/mGN node selection

The base station selects the GN as well as the mGN nodes to operate as part of the attack detection process. The GN and the mGN nodes are selected to operate with enough signal observation strength to span their respective regions of operation. The selection of the GN nodes is performed based on a uniform probability distribution. In Section 4.4, we define the mGN-Select algorithm for selection of mGN nodes for the detection scheme. The GN and mGN nodes are also pre-configured with a set of shared secret keys, to facilitate secure message communication. Each GN node is initialised with two tables in its local memory, namely, traffic flow observation table (Table 3.4) and threshold table (Figure 3.5). Each GN node stores the maximum threshold value, $t_{th_n}$, associated with each of the $r$ targets within its threshold table. The threshold values $t_{th_n}$ are computed based on Equations 3.5, 3.6 and 3.7. Once initialised, the threshold values stored in the threshold table are updated only during Phase 5 of Algorithm 4.1. The traffic flow observation table has constantly updating values depicting neighborhood traffic flow rates, towards the victim nodes, observed by the GN nodes. A comparison between corresponding values for a given target $r$ in these two tables at the end of each time epoch $\Delta_i$, decides the output signal $sig_{n}^{r}(\Delta_i)$ to be generated by each GN node $n$ for transmission to its designated mGN node, $q_n$.

Time Epoch Length Calculation

The length of the time epoch, $\Delta_{opt}$, affects the overall success in attack detection. Larger values of $\Delta_{opt}$ will delay the process of attack detection, as
the scheme will converge on a less frequent basis, effectively reducing the significance of attack detection. However, larger time epoch lengths will lead to utilisation of fewer energy resources associated with the detection process, by the detector and the mGN nodes. On the other hand, smaller values of $\Delta_{opt}$ will lead to increased energy usage by the GN and mGN nodes, albeit achieving quicker success in the attack detection process i.e. detecting an attack before significant damage is done to a target node. In Section 4.3, we formulate a tradeoff equation for computation of the $\Delta_{opt}$ value for varying application and network scenarios.

**Pattern Learning**

During this sub-phase, detector nodes are trained with patterns depicting thresholds of maximum traffic flow that is permissible for flow towards the set of $r$ selected target nodes, during a given time epoch of length $\Delta_{opt}$.

In a flat network topology (Figure 4.3), equal significance is given to each target node, since the loss of any of the nodes will incur a comparable level of damage to the network. Therefore, the threshold subpattern values for these networks are computed based on equation 3.5, reproduced as follows:

$$th_r^n = [P_r + nw(density) + \frac{1.0}{d_G(n)(r)}]$$

Higher node deployment densities will yield larger threshold values, indicating that more nodes in the network will reduce the per-node significance in such a topology. The normalised Euclidean distance of each target node, $r$, to the detector node, $n$, is also significant in defining the threshold subpattern.
values. The value of $P_r$ defines the expected number of packets by a target node during a fixed length time epoch.

Figure 4.3: Flat Network Topology with the GN Array Overlay.

In Table 4.2, we illustrate threshold patterns, constituted of individual subpatterns, for four example target nodes of Figure 4.3, generated using the above equation, for a network with $N=100$, with sensor nodes following a taxonomy, which requires the nodes to generate sensory readings once per second. Each row corresponds to a single threshold pattern for a target node $r$.

<table>
<thead>
<tr>
<th>ID($n$)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$th_{0}^0$</td>
<td>70</td>
<td>43</td>
<td>21</td>
<td>14</td>
<td>10</td>
<td>8</td>
<td>23</td>
<td>19</td>
</tr>
<tr>
<td>$th_{9}^9$</td>
<td>25</td>
<td>32</td>
<td>49</td>
<td>55</td>
<td>64</td>
<td>63</td>
<td>61</td>
<td>49</td>
</tr>
<tr>
<td>$th_{11}^{t_{11}}$</td>
<td>24</td>
<td>34</td>
<td>30</td>
<td>30</td>
<td>46</td>
<td>44</td>
<td>60</td>
<td>57</td>
</tr>
<tr>
<td>$th_{15}^{t_{15}}$</td>
<td>59</td>
<td>55</td>
<td>16</td>
<td>15</td>
<td>18</td>
<td>15</td>
<td>56</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 4.2: Flat Topology - Threshold subpatterns for target nodes $t_0$, $t_9$, $t_{11}$ and $t_{15}$, for storage within the GN nodes with ID given by ID($n$).

The proximity of each target node to the detector (GN) nodes in terms of the Euclidean distance, $d_{r_{G(n)}}^r$, play a significant role in generation of the
subpattern values. For instance, Equation 4.1 will yield a high threshold subpattern values for a target node $t_{11}$, in close proximity to the GN nodes $GN_6$ and $GN_7$, whereas the same target node distantly placed from nodes $GN_4$ and $GN_5$, will yield lower threshold subpatterns, $th_{11}^4$ and $th_{11}^5$, respectively.

In Figure 4.4, we illustrate a cluster head-based network topology, where a set of sensor nodes are designated as cluster heads, to serve as administrative and data aggregation points, for their respective clusters.

**Figure 4.4:** Cluster-based Network Topology with the GN Array Overlay.

The GN node set in a cluster-based topology store threshold subpatterns, is generated based on Equation 3.6, reproduced as follows:

$$th_d^r = [P_r + num_{ch} + \frac{1.0}{d_{Gi(d)(r)}}]$$

The density of node deployment in these networks is reflected in terms of the total number of operational clusters. Therefore, the value of $num_{ch}$, which defines the total number of cluster heads in the network, affects the values of the subpatterns generated using the above equation. Higher numbers of cluster heads in the network reduce the per-cluster head significance, and
therefore the equation yields higher threshold values, indicating more numbers of expected requests by the target nodes. The proximity of the target nodes to the detector nodes also affect the threshold subpattern value generation. In Table 4.3, we illustrate threshold patterns for two example target nodes, CH1 and CH2. The total number of nodes in each cluster is equal to 10.

<table>
<thead>
<tr>
<th>ID(n)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>th_{CH1}</td>
<td>94</td>
<td>96</td>
<td>44</td>
<td>41</td>
<td>49</td>
<td>19</td>
<td>23</td>
<td>92</td>
</tr>
<tr>
<td>th_{CH2}</td>
<td>35</td>
<td>41</td>
<td>88</td>
<td>85</td>
<td>87</td>
<td>26</td>
<td>20</td>
<td>46</td>
</tr>
</tbody>
</table>

Table 4.3: Cluster-Based Topology - Threshold subpatterns for target nodes CH1 and CH2, for storage within the GN nodes with ID given by ID(n).

As can be seen from Table 4.3, GN nodes outside the jurisdiction of a cluster head (nodes GN_2, GN_3, GN_4, GN_5, GN_6 for CH1), store low threshold values, indicating the expectation for fewer numbers of traffic packets, across cluster boundaries. However, GN nodes closer to the cluster heads store higher threshold subpatterns, to depict cluster head operation, i.e. large influx of traffic packets, within the respective cluster of operation.

In Figure 4.5, a data aggregation topology is illustrated. Considering the significance of every aggregation node in such hierarchies, each data aggregation node is considered a potential target, and has traffic flow towards it monitored by the GN nodes.

The criteria for generation of threshold subpatterns in data aggregation topologies is given in the following equation:

\[ th_d^r = [P_r + \frac{1.0}{d_G(d(r))}] \]
The significance of each data aggregation node is considered to be the same, as the loss of even a single aggregation node will make an entire arm of the network dysfunctional. Data aggregation nodes closer to the base station will expect more numbers of incoming traffic packets, and therefore higher values of $P_r$. In addition, the proximity of the data aggregation nodes to the detector nodes, defines the expected number of traffic packets towards the particular detector node, that may be tagged as legitimate network traffic flow by an observing detector node.

In Table 4.4, we illustrate the threshold patterns for two source-root data aggregation paths from Figure 4.5. The average number of incoming paths to each aggregation node is set as 3. Each source-root path has exactly four data aggregation nodes. The data aggregation nodes up the tree hierarchy, $t_{13}$ and $t_{43}$, expect higher numbers of incoming traffic packets, and therefore bear higher threshold subpatterns, whereas the data aggregation nodes close to the leaf nodes expect fewer numbers of incoming packets, and therefore lesser threshold values. The impact of the detector node proximity to the data aggregation nodes is the same as for the previous two network topologies.
4.2.2 Phase 2: Observation

In this phase, each GN node $GN_n$ observes packets initiating or transiting through its respective region of operation $S^r_n$, destined for one of the $r$ critical target nodes. A traffic observation table is defined as a table which stores the subpattern values depicting statistical features from observed traffic packets, Table 3.5. These features depict the intensity of traffic flow towards the victim node set from the region of observation of a GN node, in a given time interval $\Delta_i$. An illustration of a traffic observation table is given in Fig. 3.5. At the end of the current time epoch $\Delta_i$, the traffic observation table values are compared with corresponding subpattern values predefined and stored in the traffic threshold table, given in Table 3.1. The traffic threshold table holds a set of subpattern values depicting the maximum number of requests receivable by a given target node $r$, from the region of operation of a GN node, in a given epoch of time. Each GN node holds exactly $r$ threshold subpattern values, one each for every target node of the network.

Table 4.4: Data Aggregation Topology - Threshold subpatterns for two data aggregation paths, with target nodes: $t_4, t_5, t_9, t_{13}, t_{15}, t_{24}, t_{27}, t_{43}$ for storage within the GN nodes with ID given by $ID(n)$.

<table>
<thead>
<tr>
<th>ID($n$)</th>
<th>src-root path 0</th>
<th>src-root path 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>28 42 61 67</td>
<td>17 21 19 36</td>
</tr>
<tr>
<td>1</td>
<td>32 41 60 67</td>
<td>19 24 34 49</td>
</tr>
<tr>
<td>2</td>
<td>24 32 51 58</td>
<td>16 25 35 53</td>
</tr>
<tr>
<td>3</td>
<td>32 34 50 57</td>
<td>29 39 49 67</td>
</tr>
<tr>
<td>4</td>
<td>38 34 48 55</td>
<td>23 38 47 65</td>
</tr>
<tr>
<td>5</td>
<td>37 33 47 54</td>
<td>21 34 42 62</td>
</tr>
<tr>
<td>6</td>
<td>44 45 68 68</td>
<td>18 31 37 62</td>
</tr>
<tr>
<td>7</td>
<td>45 47 64 66</td>
<td>14 25 29 54</td>
</tr>
</tbody>
</table>
The observation process of the GN nodes continues until the end of the current time epoch is reached. The fixed lengths of time intervals of the scheme facilitate the synchronisation in the inter-node message exchange process (Phase 3). In addition, fixed length time intervals help achieve consistency in the pattern reconstruction process, required for accuracy in the detection process. Disparate time epoch lengths will lead to incomplete pattern reconstructions, thereby reducing the effectiveness of the attack detection scheme, and increasing the false alarm rates.

4.2.3 Phase 3: Communication

The completion of the observation phase is marked with the onset of the communication phase, wherein each GN node $GN_n$ communicates with exactly two other adjacent nodes, namely, the successor $(n_{succ})$, and the predecessor $(n_{pred})$ to form a dual-point linked chain of GN nodes, also referred to as the GN array. The purpose of having a dual-point linked chain is to facilitate reconstruction of complete traffic patterns from individually observed subpatterns of traffic flow. In addition, such a chain facilitates verification of peer observations by the GN nodes. Individual subpatterns are thus reconstructed to generate a complete traffic flow pattern for each of the given target nodes $r$.

All packets exchanged by the attack detection scheme are verified for authenticity using a Message Authentication Code (MAC). The tasks associated with MAC computation and verification are performed to verify both the origin as well as the integrity of all messages exchanged by the GN and
1. **Initialisation**

Selection of $n$ GN nodes

1.a **m**Select

**foreach** GN Node $n$ **do**

- Calculate $k$-Nearest Neighbors $n_k$
- Transmit $n_k$ lists to base station

end

$\{mGN_1, mGN_2, ..., mGN_m\}$ generated at base station,

$\{q_1, q_2, ..., q_n\}$ assigned to the $n$ GN nodes

1.b **Pattern Learning**

**foreach** GN Node $n$ **do**

- Generate pattern: $p_n = \{p_n^1, p_n^2, ..., p_n^r\}$

end

2. **Observation**

**foreach** GN Node $n$ **do**

**foreach** Target node $r$ **do**

- Monitor $S_{rn}$ and Update traffic observation table locally

end

end

3. **Communication**

**foreach** GN Node $n$ **do**

**foreach** Target node $r$ **do**

- if traffic observation table entry for $r > th_n^r$ then
  - Communicate with neighboring nodes $n_{succ} \land n_{pred}$ to reconstruct
  - sub-pattern $\{p_r^n, p_{nsucc}^r, p_{npred}^r\}$

end

end

**foreach** mGN Node $i$ **do**

- Cumulation of $r$ observations from $\frac{||n||}{2||m||}$ nodes in local jurisdiction, during $\Delta_i$

**foreach** Target node $r$ **do**

- Generate decision signal: $attack_r$ or $normalcy_r$

end

end

4. **Verdict**

**foreach** mGN Node $i$ **do**

- $\forall r$, if $attack_r = 1$, Transmit $attack_r$ to base station

end

5. **Pattern Update**

**foreach** GN Node $n$ **do**

- Update $th_n^r$

end

**Algorithm 4.1**: Distributed Attack Detection Scheme - Five Phases of Operation
the decision-making (mGN) nodes. Each detector (GN) node shares a pair-
wise distributed secret key with exactly three other nodes, namely, successor, predecessor and its masterGN (mGN) node, where the successor and predecessor nodes are other GN nodes, operating in the network, as part of the GN array. On the other hand, the mGN nodes will store pairwise keys that they share with each of the GN nodes in their respective jurisdiction, as well as the key that they share with the base station. The keys are scalar quantities with no direction dependance, implying: $K_{MAC}^{ab} = K_{MAC}^{ba}$. The GN nodes are pre-configured with these keys at network initialisation time by the base station.

The intent of having a shared key is to ensure the authentication and integrity checks for all GN/mGN messages, for protection against hoax packets injected by adversarial nodes in the communication channel. The computation of the MAC is performed by the UMAC algorithm proposed in (Black et al., 1999). In addition, an incremental counter value ($ctr$), as a function of the current time epoch ($\Delta_i$), is appended to all GN/mGN messages, for protection against message replay attacks. The original message $m$ is transferred un-encrypted, since the actual content of the message indicating a sub-pattern is not intelligible by the adversary, unless reconstructed completely. The reconstruction process itself will not yield much information, that may be catastrophic to the detection process. In addition, data encryption and decryption are resource consuming processes, and therefore must be rarely used on sensor nodes. All messages exchanged between GN/mGN nodes and the base station, will have the following format:

$$\{\forall a \mid a \in \{senders\} \land \forall b \mid b \in \{receivers\}\}$$

where,
\{\text{GN source node} \in a \land n\text{pred}(a) \in b\} \lor \{\text{GN source node} \in a \land n\text{succ}(a) \in b\} \lor \{\text{GN source node} \in a \land Q(a) \in b\} \lor \{\text{mGN source node} \in a \land \text{base station} \in b\},

\text{a} \rightarrow \text{b}: \{m\}, MAC(K_{MAC}^{ab}, m, ctr(\Delta_i))

where,

\begin{align*}
m &= \text{Message for transfer} \\
ctr(\Delta_i) &= \text{counter value as a function of the current time epoch } \Delta_i \\
K_{mac}^{ab} &: \text{MAC computation key shared between } a \text{ and } b
\end{align*}

The MAC is nothing more than a series of operations on the plain text message with the output of the MAC operation being much smaller in size as compared to the size of the input. The MAC is typically computed using a one way hash function, where the resultant output is irreversible.

The counter value, $ctr_{\Delta_i}$, is incremental, since it is based on the current time stamp (including date), which is a one-way increasing function. The counter ensures message order verification at the receiver end. Therefore, all communicated messages between the detector nodes and the base station are protected against message replay attacks.

Each GN node holds exactly two keys for securely communicating with its peer GN nodes, and exactly one key to communicate with its designated mGN node. In addition, each mGN node in the network holds a set of keys to communicate with each of its designated GN nodes, and a single key to communicate with the base station. Considering the symmetric nature of the keys, where $K_{MAC}^{ab} = K_{MAC}^{ba}$, duplication on keys shared between the mGN
nodes and the GN nodes is avoided. The total number of keys in the network are therefore given by: $3n + q$, where $n$ is the total number of detector (GN) nodes, and $q$ is the total number of mGN nodes in the network.

At each GN node, if the number of incoming requests for a particular target $r$ during the current time epoch exceed the stored threshold $th^r_n$ value in the pattern table, and its successor and predecessor nodes have also detected similar anomalies given by their respective sub-patterns, $p^r_{nsucc}$ and $p^r_{npred}$, the GN node $n$ will generate an attack$_r$ signal in the current time epoch. On the contrary, a normalcy$_r$ signal generated by the GN nodes implies incomplete or no-match between the observed traffic pattern and the stored pattern of anomalous behavior for traffic destined for node $r$.

All GN node communication takes place in parallel, and therefore, the overall communication delay incurred is minimal (Section 4.5). After comparison of the subpattern values with the adjacent GN nodes, the outcomes of the pattern recognition process from alternating members of the GN array are communicated to their designated mGN nodes. Due to the alternating communication process between adjacent GN nodes of the GN array, the overall communication overhead, in terms of the energy use associated with the scheme, is thus halved.

4.2.4 Phase 4: Verdict

Neighbouring GN nodes alternate in communicating with their designated mGN nodes, in consecutive time epochs, to avoid duplication in the messages sent to the mGN nodes. Therefore, if during time epoch $\Delta_{i-1}$, GN nodes $G_{n-1}$ and $G_{n+1}$ communicate with $M_{G(n-1)}$ and $M_{G(n+1)}$, respectively, then
during $\Delta_i$, nodes $G_{n-1}$ and $G_{n+1}$ will enter sleep mode, and node $G_n$ will perform the communication.

The mGN nodes are selected as such that the connectivity of each of the GN nodes to an mGN node in the network is possible. If distantly placed GN nodes in a network spanning a large geographical area are selected to operate as mGN nodes, the GN-mGN node interconnectivity will not be achieved entirely. The missing links in the network between certain GN and mGN nodes will reflect on the performance of the detection scheme. In particular, the false alarm rates of the scheme will be increased due to the dependance of the mGN nodes on default decision messages, in the absence of actual observation results from inaccessible GN nodes. We define the $m$Select algorithm in Section 4.4 for selection of a subset of GN nodes to act as mGN nodes in a network with given parameters, so as to achieve 100% connectivity between the GN and the mGN nodes.

Each mGN node expects exactly $\frac{||n||_2}{2||m||}$ GN nodes in their respective local jurisdictions, to send them a boolean-valued signal for each of the $r$ targets to confirm an attack. At the end of the current time epoch $\Delta_i$, if the number of GN $\text{attack}$ signals for any or all of the specified targets arriving at the mGN nodes equals to half of the number of participating GN nodes $\frac{n}{2}$, the traffic flow is classified as an attack: $\forall r$, if $\prod_{i=1}^{\frac{n}{2}} \text{attack}_r(i) = 1 \Rightarrow$ an attack against $r$ is in progress. If the aggregate number of arriving $\text{attack}$ signals at all mGN nodes is less than $\frac{n}{2}$, a normalcy signal is generated by the detection scheme.
4.2.5 Phase 5: Pattern Update

The decaying energy contents of individual sensor nodes in the network demand the need for constant update of pattern values stored in the Pattern Tables of GN nodes. The accuracy of the pattern recognition scheme depends on the frequency of update of the $th_{t_0}$ values. If the update rate is not at pace with the rate of declining energy resources of the target nodes (energy consumption rates), incoming attack traffic may lead to exhaustion of energy of the target nodes, and remain unnoticed by the observing GN nodes.

During this phase, the pattern values are updated based on one of two approaches: 

a) Expected traffic inflow governing equations, and

b) Actual traffic inflow-based pattern update. An analysis and comparison of the two has been done in the next chapter.

Upon successful confirmation of an attack signal, the base station sends a signal to induce node $r$ into sleep mode for a finite period of time. Subsequently the base station ensures that if alternative resources are available, they are sent a signal to designate them the task of continuing with the sensing operations from the region of operation of node $r$. For instance, if node $r$ belonged to a DA-based topology responsible for aggregation of received data, the base station sends a request to another active node available within the vicinity of node $r$ instructing it to take over the data aggregation responsibilities of $r$.

In Sections 5.2.3 and 5.2.5, we study the attack detection rates and the false alarm rates of the detection scheme, under variations in the network size ($N$), detector node ratio ($n$) and the attack traffic intensity. In addition, in
Section 5.2.4, we analyse the error rate in detection associated with variations of the pattern update frequencies of the attack detection scheme.

4.3 Computation of the Optimal Time Epoch Length \( (\Delta_{opt}) \)

The length of \( \Delta_{opt} \) has a significant impact on several other factors, such as the effect of attack detection, false alarm rate and the energy consumption rate associated with the attack detection scheme. In this section, we formulate an equation to tradeoff between frequent attack detection and detector/mGN node energy resource consumption. Higher frequency of detection scheme convergence will lead to higher energy consumption rates in the detector and mGN nodes. However, such an approach will help detect an attack before significant loss is incurred on a target node. Smaller frequencies of convergence of the detection scheme would lead to conservation of the energy resources of the GN and mGN nodes, at the cost of lesser effectiveness of the detection scheme, associated with delayed detection of an attack.

All GN and mGN nodes communicate with each other, once, during Phase 3 of the attack detection process. The frequency at which the GN and the mGN nodes communicate with each other depends on the selected length of the time epoch, \( \Delta_{opt} \). Large values of \( \Delta_{opt} \) will cause the attack detection scheme to converge on a less frequent basis:

**Definition 4.1.** Let \( l \) be a set of packets, \( \{pk_1, pk_2, ..., pk_l\} \), launched by an adversary towards a potential target \( r \), during a time epoch \( \Delta_i \). Let \( ts_l \) be the time stamp (initialisation time) of \( pk_l \), then if \( \text{length}(\Delta_i) \) is large and
\{\forall t, ts_l < (\Delta_i - \Delta_{i-1})\}, the attack will inflict damage on node r before being detected.

Similarly, if the time epoch length is small, the attack detection scheme will converge on a more frequent basis. Such a situation will lead to rapid exhaustion of the energy resources in the GN/mGN nodes owing to higher frequencies of inter-GN node communications.

We define an analytical model to determine the optimal length of the time epoch, \( \Delta_{opt} \), for a given network with given values for its parameters. The total number of mGN nodes in the network is given by \( m \), and the total number of detector nodes is given by \( n \). The average number of detector nodes per each mGN node is given by: \( \frac{n}{m} \). The actual value of \( m \) is computed using the mGN computation algorithm (Algorithm 4.2). The average distances between the mGN and the detector nodes, \( d_{mn} \), and the average distance between the mGN nodes and the base station, \( d_{mb} \), are calculated for the different network types, through experiments.

We consider the total cost of the detection scheme, \( Cost_{total} \), as a summation of two costs - \( C1 \) and \( C2 \):

\[
Cost_{total} = C1 + C2 \tag{4.1}
\]

where,

\( C1 \) = Cost associated with energy consumption by the detection scheme i.e. resource usage by the detector nodes/mGN nodes.

\( C2 \) = Cost incurred on the network due to node loss because of an attack.

The total energy usage of the scheme can be modeled as follows:
\[ E_{\text{total}} = m \cdot [E_{mgn} + \frac{E_{gn}}{2} \cdot (\frac{n}{m} - 1)] \]  

(4.2)

where, \( E_{mgn} \) is the energy consumed by the mGN nodes, and \( E_{gn} \) is the energy consumed by the \( \frac{n}{m} \) detector nodes in the network. Considering the relatively shorter communication distances between the mGN nodes and the detector nodes, we model the power loss on the GN-mGN channel as a free-space model (Kim et al., 2005), wherein the power loss is the square of the inter-node distance \( d_{mn}^2 \). The mGN to base station distances are expected to be longer, and therefore the multi-path fading model (Kim et al., 2005), is used for power-loss modeling on the mGN-base station channel. In this model, the power loss is defined as the fourth power of the channel distance \( (d_{mb}^4) \).

The energy usage of the mGN nodes, given by \( E_{mgn} \), depends on the rate at which messages are received from their respective GN nodes. This rate in turn depends on the frequency of convergence of the detection scheme. Larger values of \( \Delta_{\text{opt}} \) will lead to higher energy drop rates for the mGN nodes, and therefore the \( \Delta_{\text{opt}} \) is multiplied with the energy in equation 4.2. The value of \( E_{mgn} \) is a function of energies utilised for the following processes: receiving data from the \( \frac{n}{m} - 1 \) nodes within an mGN node’s jurisdiction, aggregation of data from these nodes \( (E_{DA}) \), and subsequent transmission of data to the base station \( (E_{mb} = E_{\text{elec}} + \epsilon_{mp}(d_{mb}^4)) \).

\( \epsilon_{mp} \) is the energy utilized for transmission of a bit of message over a given distance using the multi-path fading model. \( E_{\text{elec}} \) is the electronics energy, associated with operations such as digital coding, modulation, filtering, and
spreading of the signal. The energy consumed by an mGN node for an \( l \)-bit message is given by:

\[
E_{mgn} = l \left[ \left( \frac{n}{m} - 1 \right) E_{elec} + \left( \frac{n}{m} - 1 \right) E_{DA} + E_{elec} + \epsilon_{mp}(d_{mb}^4) \right]
\] (4.3)

The free-space power loss model for the mGN-GN communication channel implies the total energy is proportional to the square of the mGN-GN distance, \( d_{mn}^2 \), and the total energy use by the GN nodes is given by:

\[
E_{gn} = l \left[ E_{elec} + \epsilon_{fs}(d_{mn}^2) \right]
\] (4.4)

where, \( \epsilon_{fs} \) is defined as the energy utilized per bit of message transfer using the free-space model.

The cost of energy consumption is inversely proportional to the value of \( \Delta_{opt} \). Therefore, the total cost \( C1 \) is given by \( \frac{E_{total}}{\Delta_{opt}} \). Cost \( C1 \) can be derived from Equations 4.2, 4.3 and 4.4 as follows:

\[
C1 = \frac{m.l}{2.\Delta_{opt}} \left[ \left( \frac{n - m}{m} \right) E_{elec} + \left( \frac{n - m}{m} \right) E_{DA} + E_{elec} + \epsilon_{mp}.d_{mb}^4 \right] + \frac{l.(n - m)}{2.\Delta_{opt}} \left[ E_{elec} + \epsilon_{fs}.d_{mn}^2 \right]
\] (4.5)

The cost incurred due to node loss, \( C2 \), is directly proportional to the length of the time epoch, and is therefore given by:

\[
C2 = \Delta_{opt}.\alpha.TI_e
\] (4.6)
The factor \( \alpha \), defined as the application aspect value is inversely proportional to the value of \( \Delta_{\text{opt}} \), and normalised between \( \{0.0 - 1.0\} \). Certain applications of sensor networks signify the need for having consistently high attack detection rates, with less regard to the added cost of more energy resource usage by the detector/decision-making nodes. In such scenarios, higher values of \( \alpha \), close to unity are considered. On the other hand, other applications willingly compromise the success in the attack detection process, by requiring the detection scheme to converge on a less frequent basis, so as to reduce the overall energy consumption rates appertaining to the more frequent convergence of the GN/mGN nodes, in turn increasing the overall longevity of the sensor network. In this case, smaller values of \( \alpha \) help reduce the overall energy consumption rates incurred by the attack detection scheme. The value of \( \alpha \) is selected at network initialisation time, and it affects the corresponding value of the parameter \( k \) of the \( mSelect \) algorithm, as will be elaborated upon in Section 4.4.

The overall cost of the attack detection scheme is therefore given by:

\[
\text{Cost}_{\text{total}} = \frac{l.(n - m).E_{\text{elec}}}{2.\Delta_{\text{opt}}} + \frac{l.(n - m).E_{\text{DA}}}{2.\Delta_{\text{opt}}} + \frac{m.l.E_{\text{elec}}}{2.\Delta_{\text{opt}}} + \frac{m.l.\epsilon_{mp}d_{mb}^4}{2.\Delta_{\text{opt}}} + \frac{l.(n - m).E_{\text{elec}}}{2.\Delta_{\text{opt}}} + \frac{l.(n - m).\epsilon_{fs}d_{mn}^2}{\Delta_{\text{opt}}} + \Delta_{\text{opt}}.\alpha.TI_e \tag{4.7}
\]

The optimal length of a time epoch, \( \Delta_{\text{opt}} \), is given by the solution of the first derivative of Equation 4.7 (by equating the derivative to 0):

\[
\Delta_{\text{opt}} = \sqrt{2.(n - m).[E_{\text{elec}} + \frac{E_{DA}}{2} + \epsilon_{fs}d_{mn}^2] + \epsilon_{mp}d_{mb}^4} \div \alpha.TI_e \tag{4.8}
\]
where,
\[ \Delta_{opt} = \text{Optimal length of time epoch.} \]
\[ \alpha = \text{application aspect value \{0.0 - 1.0\}.} \]
\[ N = \text{Number of sensor nodes.} \]
\[ n = \text{Number of detector nodes.} \]
\[ m = \text{Number of mGN nodes.} \]
\[ TI_e = \text{Expected Traffic Intensity (nJ/bit).} \]
\[ d_{mn} = \text{Average distance from detector node to mGN node (experimental).} \]
\[ d_{mb} = \text{Average distance from mGN node to base station (experimental).} \]
\[ E_{DA} = \text{Data aggregation energy} = 5 \text{ nJ/bit/signal.} \]
\[ E_{elec} = \text{Hardware energy} = 50 \text{ nJ/bit.} \]
\[ \epsilon_{fs}=10 \text{ pJ/bit/m}^2. \]
\[ \epsilon_{mp}=0.0013 \text{ pJ/bit/m}^2. \]

The traffic intensity \( TI \) is defined as the total number of traffic packets destined for a target node per unit of time. For each arriving packet at the target node, the energy resource usage for the processing of the packet is given by \( E_{DA} = 50 \text{ nJ/bit.} \) Therefore, for an estimated 2 byte attack packet, the total energy utilised by the target node for its processing is given by: 800 nJ. During a given time epoch of length \( \Delta_{opt} \), a \( TI \) value of 500 (i.e. 500 pkts/\( \Delta \)), yields a total energy usage of 800 \( \mu \)J.

In Section 5.2.2, we perform a quantitative study of the impact of the value of \( \Delta_{opt} \) on the energy consumption rates of the GN, mGN, and the target nodes undergoing a distributed flooding attack.
4.4 Selection of the Decision-Making (mGN) Nodes

The GN-layer of the attack detection scheme is responsible for the observation, monitoring and reporting of traffic observation subpatterns to their designated mGN nodes at the end of defined epochs of time. The mGN nodes are responsible for taking a decision marking an attack in progress, or normalcy in traffic flow towards the target nodes.

The mGN nodes of the scheme communicate with the base station at the end of each time epoch. As a result of having the mGN nodes operate as an intermediary between the GN nodes and the base station, the communication distances to be covered by the GN decision messages are reduced significantly. The mGN nodes also ensure that localised monitoring of GN nodes is attained. Dead/Inactive GN nodes are observed and reported by their designated mGN nodes to the base station. An attack is signalled as an event which involves each mGN node post-coordination with their respective GN nodes, generating an attack signal for a particular target node \( r \) during the current time epoch.

In a non-mGN scenario, the individual GN nodes need to communicate with the base station at the end of each time interval for purposes of conveying their local traffic observations. The longer communication distances in such a scenario imply quicker exhaustion of the energy resources of individual GN nodes.

A large number of mGN nodes operating in the network will have an impact on the delays associated with the completion of all phases of operation of the detection scheme. This delay will in turn affect the accuracy in attack detection, and reduce the possibility of applying mitigation techniques, upon
successful detection of an attack. In addition, a significant overhead is in-
curred on the mGN nodes, for receiving and transmitting messages. Having
very few operational mGN nodes in the network will increase the per-node
overhead on these fewer number of nodes, and will lead to the rapid exhaus-
tion of only a select few nodes of the network. It is therefore inferrable that
the lesser the number of mGN nodes, the longer the durability of the attack
detection scheme. In this section, we propose an algorithm for selection of a
minimal number of mGN nodes, based on the following two criteria - a) total
number of GN nodes in the network, and b) the communication range of each
node.

Several algorithms for neighbour-based topology control have been pro-
posed in the literature. The $k$-Neigh protocol (Blough et al., 2003a)(Blough
et al., 2003b)(Santi, 2005), is a topology control protocol for Wireless and
AdHoc networks, for generation of the $k$-closest neighbor lists within each
participating mobile or sensor node, based on node transmission ranges and
inter-node distances. In (Wattenhofer and Zollinger, 2004), a communication
link quality-based topology control algorithm is proposed, for generation of
closest neighbour lists within the wireless nodes. Both these protocols assume
varying node transmission ranges based on the density of node deployment
($N$).

We define the $m$Select algorithm for generation of the mGN node set, for
fixed transmission sensor nodes, wherein the transmission range is considered
to be 50 meters (Krohn et al., 2006). The algorithm is executed during
the Phase 1 of the attack detection scheme. It generates the mGN node
list based on the proximity of the detector nodes to each other, in terms of
communication accessibility, and runs with the following assumptions:
1. All nodes post-deployment are stationary.

2. All nodes have communication access to the base station.

3. The maximum transmission range within each of the nodes is the same.

4. Each node, upon message exchange, has an estimate of the Euclidean distance to every other node.

5. The algorithm is initiated by the base station during the Initialisation phase of the Algorithm 4.1.

During the initialisation phase of the detection scheme, all GN nodes broadcast their IDs with maximum power. Each node upon receiving the broadcast messages creates a list of $k$ closest neighbors based on its relative Euclidean distance from the source nodes. Let $n$ denote the total number of GN nodes in the network. A directed graph representation of all GN nodes can be defined as follows:

**Definition 4.2.** $Z_k=(G, E_k)$ is a directed graph with $|G|$ = number of GN nodes in the network, and $E_k$ = set of $G.k$ edges, connecting nodes belonging to $G$ with their $k$ nearest neighbors.

A symmetric $k$-neighbor sub-graph of a directed graph $Z_k$, is therefore given by:

**Definition 4.3.** A symmetric sub-graph for a given directed graph $Z_k$ is defined as an undirected graph $Z_k^- = (N, E_k^-)$, where the undirected edge $(u, v) \in E_k^-$ if and only if $(u, v) \in E_k$ and $(v, u) \in E_k$.

The value of $k$ defines the probability of connectivity of the graph $Z_k^-$, i.e. the probability of having a complete $k$-nearest neighbor list constructed at
the end of the initialisation phase within each of the $n$ GN nodes, assuming that the maximum transmission range of each sensor node is the same. In both (Blough et al., 2003a) and (Wattenhofer and Zollinger, 2004), the value of $k$ is varied, and an empirical study is performed, to define the optimal $k$ value. In our algorithm, the value of $k$ is inversely dependant on the value of the application aspect factor $\alpha$. The maximum value of $k$ is equal to $N-1$, implying that each node can generate a local list of $k$-accessible neighbours, that may span the entire network. Similarly, the lowest value of $k$ is unity, implying that each node will have a single neighbour. Applications demanding significance in the accuracy in attack detection over the energy consumption rates of the GN/mGN nodes will set a higher value for $\alpha$, leading to lower values for $k$. Similarly, other applications signifying the need for conserving the energy content of the GN/mGN nodes at the cost of lesser accuracies in attack detection rates will set a lower value of $\alpha$, leading to higher values for $k$.

Upon completion of the broadcast phase, each node $i$ computes a local list consisting of $k$ closest neighbors. The nodes subsequently broadcast their respective $k$ lists at maximum power. This second broadcast round facilitates the computation of symmetric neighbors within the $k$ lists of the sensor nodes i.e. if $j \in k\text{list}_i \Rightarrow i \in k\text{list}_j$. After the second broadcast round is complete, a list of $k$-closest neighbors is defined and stored locally within each node of the network. Subsequently, for purposes of topology control, the transmit power of each node is set to the minimum power required by the node for transmitting a message to its farthest ($k^{th}$) neighbor.

In Algorithm 4.2, we illustrate the steps of execution of the $m$Select algorithm by the GN nodes. The algorithm provisions for steps required to
generate a minimum-sized set of common neighbors within each of the GN nodes, from their locally generated symmetric $k$-neighbour lists. The resultant set, notated as $M$, is the set of mGN nodes for the attack detection scheme.

The set $M$ is defined as the intersection of common neighbors from the $n.k$ lists transmitted by the GN nodes to the base station at network initialisation time. $\|M\| = O(k)$, implying that the size of the set will never exceed the value of $k$.

**Definition 4.4.** $M = \{mGN_1, mGN_2, ..., mGN_m \mid m \in \mathbb{N} \land m \ll n\}$

At the end of the Phase 1 of Algorithm 4.1, each GN node will have a designated mGN node given by: $q_n$. The selected value of $k$ affects the size of the mGN node set:

**Corollary 4.1.** $\|M\| \propto \frac{1}{k}$, large values of $k$ will lead to smaller $\|M\|$, and vice versa.

**Proof.** Let $J_n$ be the set of $k$ neighbors within each of the GN nodes $n$, at initialisation time. If value of $k$ is small, then for any two GN nodes $GN_x, GN_y$, the probability of connectivity $P(a \in J_x \land b \in J_y) \approx 0$. On the contrary, for larger values of $k$, $P(a \in J_x \land b \in J_y) = 1$.

The larger the value of $k$, the smaller the cardinality of the $M$ set will be. However, the value of $k$ will vary on a per-node basis, depending on the placement of each node. In Section 5.2.2, we study the energy consumption rates of both the GN as well as the mGN nodes of the detection scheme, for variations in the value of the application aspect value $\alpha$. 

134
Input: List of all GN nodes in the network, \( k \)

Output: Designation of an mGN node for each GN node \( n \): 
\[
\{q_1, q_2, ..., q_n\}
\]

Generation of \( n.k \) lists of \( k \)-closest neighbors, one in each GN node:

foreach GN node \( i \in G \) do
    Generate broadcast message = \{ coordinates(i), cluster number(i),
    current energy content(i) \}
end

foreach node \( i \in G \) do
    foreach Message \( mr(n) \) received do
        insert coordinates of node \( n \) in ordered \( k \)-list(i) based on dist(i, n), energy content(n)
    end
end

Neighbour list exchange:

foreach GN node \( i \in G \) do
    Generate \( k \)-list(i) and broadcast at maximum power
end

\( k \)-list generation:

foreach GN node \( i \in G \) do
    Generate symmetric neighbour list \( k \)-list(i)
end

\( k \)-list exchange:

foreach GN node \( i \in G \) do
    Transmit \( k \)-list(i) to base station
end

At base station:
Calculate minimum cover set over all sorted lists \( k \)-list(\( G_1, G_2, ..., G_n \))

foreach GN node \( i \in G \) do
    \( min\text{-}distance_i \) = Large Integer;
    foreach mGN node \( m \) do
        if Euclidean(\( m, i \)) < \( min\text{-}distance_i \) then
            \( q_i = m \);
        end
    end
end

Algorithm 4.2: Steps of execution of the mSelect Algorithm.
4.5 Efficiency Analysis

In this section we analyse the overhead incurred on a sensor node participating in the attack detection process. On an average, each GN node stores 1Byte of subpatterns for each of the \( r \) target nodes. For a network with 1024 nodes, with 50\% of nodes being targets, each GN node \( GN_n \) will have to store approximately 500B of sub-patterns, which is less than 6\% of a typical Mica’s memory (Perrig and Tygar, 2002). Each \( GN_n \) will exchange exactly 2 packets with its adjacent GN nodes \( nsucc_n \) and \( npred_n \), and a single packet with its mGN node \( q_n \), at the end of each \( \Delta_i \). Therefore, the Communication Cost for the scheme is given by \( O(n) \).

Our scheme enforces strict rules for the exchange of subpatterns at the end of each time epoch. Selective and intelligent exchange of subpatterns between the GN nodes can help reduce the overall communication overhead, but at the cost of a markable increase in the number of false alarms. In this case, fewer messages exchanged between the operating GN nodes would imply lesser likelihood of detecting GN node compromise and failure.

**Lemma 4.1.** The convergence delay for the attack detection scheme at the end of a given time epoch, \( \Delta_i \) is given by: 
\[
(3.d1 + \frac{n}{m}.d2 + m.d3)
\]

where,
- \( d1 = \text{Average GN to GN communication delay} \)
- \( d2 = \text{Average GN to mGN communication delay} \)
- \( d3 = \text{Average mGN to BS communication delay} \)

**Proof.** Consider the network of GN nodes to be a graph given by \( G = (V, E) \), with \( V = \{g_1, g_2, g_3, ..., g_n\} \), and \( E = \{\text{Set of graph edges}\} \). Let \( e = n \). In the best case scenario, all possible inter-GN node communications are
achievable in parallel. Adjacent nodes may either transmit or receive at any given instance of time. Let the set $S_e =$ \{set of neighbors of node $e$\}. The problem may be formulated as a graph coloring problem, with the task of finding the minimum number of colors required to color the graph of GN nodes such that for any two GN nodes, $g_1$ and $g_2$, assigned the same color,

$$S_{g_1} \cap S_{g_2} = \emptyset,$$

where $\{g_1, g_2\} \in V$.

From observation, we can see that if every fourth GN node of the GN array communicates with its adjacent neighbors at the same instance of time, the best case parallelism in the GN node communication phase can be achieved. For instance, if GN nodes $g_1$ and $g_2$ communicate with their respective neighbors ($g_2, g_1 \land g_3$) at the same instance of time, one or both the communication messages may be lost due to the atomicity of the sensor transmit/receive operations. Similarly, if GN nodes $g_1$ and $g_3$ communicate with their respective neighbors ($g_2, g_2 \land g_4$) at the same time instance, again a high chance of communication message loss exists. However, if every fourth node communicates with its neighbors at the same time instance: E.g. GN node $g_1$ with neighbor $g_2$, and node $g_4$ with neighbors $g_3$ and $g_5$, the probability of a message loss due to collision on a shared communication channel is 0. Therefore, an average wait of 3 cycles is required for the GN node-node communication to be completed, and therefore, the convergence delay is given by: $3d_1 + n.d_2 + m.d_3$.

The aggregation of localised attack detection results by the mGN nodes imply that the total number of messages exchanged is equal to the summation of the number of GN nodes operational in the network and the total number of mGN nodes.
Lemma 4.2. Total communication signals exchanged by the detection scheme in a single time epoch, $\Delta$, is given by: $O(2n + \frac{n^2}{2} + m)$

Proof. Each of the $n$ active GN nodes in the network participate in exchange of their local observations with at most two other GN nodes. Upon completion of the observation signal exchange phase, the $\frac{n^2}{2}$ GN nodes communicate with their designated mGN nodes. The mGN nodes subsequently communicate with the base station, thus generating an additional $m$ signals for transmission. Therefore, the total number of communication messages transmitted during Phase 3 of Algorithm 4.1, in a given time epoch $\Delta_i$, is at most $2n + \frac{n^2}{2} + m$.

4.6 Conclusions

In this chapter, we proposed a distributed pattern recognition scheme for detecting DDoS attack patterns in wireless sensor networks. The attack detection scheme consists of five phases of operation, to be executed sequentially within each epoch of time, of length=$\Delta_{opt}$. During the Initialisation phase, the detector nodes are selected by the base station. Distinct topology-based threshold patterns for each of the $r$ target nodes in the network are generated for comparison with actual traffic flow observations by the attack detector (GN) nodes. Subpatterns of threshold values depicting distributed flooding attacks against the target node set $T$ are generated using Equations 3.5, 3.6 and 3.7, defined in Chapter 3, and stored in each of the GN nodes respectively. The base station is also responsible for selection of the mGN nodes during the Initialisation phase of the attack detection process. The $m$Select algorithm has been proposed by us to select the mGN nodes in the network,
based on network connectivity and GN node deployment densities. The proposed algorithm ensures that the set of mGN nodes selected is the smallest required, so as to reduce the overhead incurred on the network due to a large number of operating mGN nodes. During Phase 2: Observation phase of the scheme, the attack detector nodes monitor and update their individual traffic flow tables with traffic observation readings from the network. In Phase 3: Communication phase, the observed readings in the form of sub-patterns, are exchanged with peer GN nodes in the network, for verification and pattern reconstruction purposes. During Phase 4: Verdict, the final verdict on whether an attack is in progress, is taken by each of the mGN nodes of the network, and communicated to the base station. During Phase 5 (Pattern Update) of the scheme, individual subpatterns for each of the $r$ target nodes of the network are updated to depict accurate energy content values of the target nodes in terms of the numbers of traffic packets receivable by them in a given epoch of time.

We formulated a tradeoff equation to compute the optimal length of a time epoch, $\Delta_{opt}$ for the scheme to converge in, so as to achieve reasonable attack detection rates at the cost of minimal energy resource usage by the detector and the mGN nodes. Certain applications of wireless sensor networks require the scheme to converge less frequently so as to reduce the overhead, and increase the lifetimes of the network, at the cost of lower attack detection rates. On the other hand, other applications require the scheme to converge more frequently to increase the attack detection rates, at the cost of more resource usage. We incorporate both scenarios within the tradeoff formulation for the optimal time window.
In Chapter 5, we provide a performance evaluation of our proposed attack detection scheme, for variations in the following algorithmic and network-level parameters:

- Number of detector nodes \((n)\).
- Network traffic intensities (adversarial nodes).
- Node deployment densities.

The purpose of the simulation experiments is to quantify the performance of our scheme based on the following metrics:

- Attack detection rates.
- False positive rates.
- False negative rates.
- GN and mGN node energy decay rates.

We also compare the obtained results with corresponding results from a Self Organising Map-based centralised attack detection approach, to prove the superiority of our distributed attack detection over the centralised technique.
Chapter 5

Performance Analysis and Benchmarking

In this chapter, we evaluate the performance of our proposed distributed attack detection scheme, through experiments and simulation analysis. The scheme proposed in Chapter 4, performs distributed denial of service attack detection, in the presence of injected nodes, inclusive of laptop-class nodes, in the network. As part of the detection process, detector nodes monitor network traffic flow towards a set of victim nodes, and further, reconstruct patterns of observed network traffic, to facilitate attack decision making. The performance of the scheme is affected by several algorithmic as well as network parameters, that need to be defined, at network initialisation time. We study the effect of variation of these algorithmic and network-level parameters on the outcomes of the proposed attack detection scheme. We quantify the results obtained for variations in these parameter values, based on simulation experiments. The metrics that are compared and analysed as part of the
simulation experiment are: attack detection rate, false positive rate, false negative rate and detector/mGN node energy utilisation rates.

We also perform a comparative analysis of the acquired experimental results for the proposed scheme, with corresponding results obtained from the simulation of a centralised Self Organising Map-based attack detection scheme. The purpose of this comparison is to establish the superiority of the proposed distributed pattern recognition approach over other centralised techniques for detection of distributed denial of service attacks in wireless sensor networks.

5.1 Introduction

The distributed denial of service attack detection scheme proposed in Chapter 4, performs attack detection in the presence of injected adversarial nodes. The intensity of the attack traffic increases with corresponding increase in the total number of adversarial nodes in the network, assuming participation of all such nodes in the attack process. If the adversarial nodes are placed at multiple locations in the network, the traffic intensity will increase from several ends of the network, thus comprising a distributed denial of service attack. The algorithmic parameters associated with the attack detection algorithm are: optimal time epoch length ($\Delta_{opt}$), number of detector GN nodes $n$, and the pattern update rates for each of the $r$ target nodes. The network-level parameters that affect the outcomes of the detection scheme are: node deployment densities and initial node energy contents.
We analyse the outcomes of the attack detection process in terms of the total number of attacks detected successfully, for variations in the above-defined parameter values. The false alarms generated by the scheme are categorised into false positives and false negatives. We study the effect of the variation in parameter values on the false alarm rates of the scheme. The overall effect of the attack detection through a SOM-based centralised mechanism is compared with results acquired for our proposed scheme, under variations of the parameter values. The evaluation of the scheme is quantified based on the following metrics:

- Attack detection rates.
- False positive rates.
- False negative rates.
- Energy decay rates.

Higher detection rates imply quicker response times by the base station in replacement or reallocation of the victim node tasks to other nodes. Lower detection rates imply rapid exhaustion of the energy contents within the target nodes, thus reducing the overall functionality of the network.

The false alarm rates of the scheme are also studied to analyse the shortcomings of the algorithm, under variations in the parameter values. False positive rates imply the incorrect classification of legitimate network packets as attack packets by the detection scheme. Higher values of false positive rates will lead to the incorrect reallocation and/or replacement of sensor nodes, assumed to be under an attack. False negative rates are defined as the rate at which the detection scheme classifies malicious packets as legitimate packets.
Higher false negative rates imply that the detection rate has faltered in its task of accurately detecting attacks, and therefore will lead to higher success, in terms of rapid energy exhaustion of the victim nodes, by the attacker nodes.

The energy decay rate of the individual sensor nodes provide an estimate on the expected lifetimes of the nodes. Higher energy decay rates will lead to rapid decline in the network resources, and reduce the overall lifetime of the network. Considering the inaccessibility of most sensor networks post-deployment, it is very crucial to safeguard the limited on-node energy resources. We therefore analyse the effect of the detection process on the energy content of the GN as well as the mGN nodes of the network, to derive results justifying an ideal ratio of these nodes, to facilitate scheme operation with minimal energy overhead, without compromising the accuracy in attack detection.

We define the initial parameter values to be considered for all experimental analysis, in Section 5.2.1. Our initial experiments are performed to study the attack detection rates for variations in the network traffic intensities, network dimensions, as well as the node deployment densities. The purpose of this evaluation is to study the effectiveness of the detection scheme in the presence of varying numbers of attacker nodes, under different sensor network-application scenarios. In Section 5.2.2, we study the energy decay rates of the GN, mGN, and the target nodes for varying applications of the sensor networks (varying $\alpha$ and $\Delta_{opt}$). In Section 5.2.3, we study the attack detection rates of the scheme for variations in several parameters. In Section 5.2.4, we compare the effect of the threshold subpattern update rate on the detection process, based on expected network traffic, as well as the actual
observed traffic. Subsequently, in Section 5.2.5, we analyse the false alarm rates of the detection scheme, for variations in the network parameter values, and the attack traffic intensities.

In Section 5.3, we analyse the experimental results generated from a SOM-based, centralised approach for detection of distributed denial of service attack patterns. The outcomes of this experiment were studied to prove the need for having a distributed pattern recognition mechanism in place to detect distributed denial of service attacks in wireless sensor networks, which is achieved through the distributed detection scheme of Chapter 4. In Section 5.4, we compare the performance of the two approaches, in terms of the metrics defined above.

The contributions of this chapter are as follows:

• We measure the performance of our scheme based on the following metrics:
  – Attack detection rates.
  – False positive rates.
  – False negative rates.
  – GN and mGN node energy decay rates.

• We test the effectiveness of the proposed scheme for variations in the following algorithmic and network-level parameters:
  – Number of detector nodes (n).
  – Network traffic intensities (adversarial nodes).
  – Node deployment densities.
• We prove the superiority of our proposed distributed pattern recognition scheme over a centralised Self Organising Map-based approach, for variations in all the above parameter values.

5.2 Analysis

5.2.1 Experimental Setup

Wireless sensor networks are deployed for specific sensing and reporting applications. The area of sensor node deployment depends on the nature of the application, and the total number of nodes to be deployed depend on several characteristics of the application, namely, expected node lifetimes, expected per-node load and node sensing ranges. Generally, most networks studied span a two dimensional area of size 100m x 100m (Fang et al., 2003)(Du et al., 2005). In (Ding and Liu, 2004), a 200m x 200m network is considered to study a centralised data gathering and communication mechanism, based on an ant colony optimisation algorithm. In (Levis et al., 2004), a 50m x 50m area is simulated to study a novel algorithm for code propagation in wireless sensor networks. All the above networks are considered with variations in the total numbers of nodes deployed. The average number of nodes in the network depends on the communication range of individual sensor nodes, as well as the topological specifications of the network. For large dimension networks, either sensor nodes with strong communication antennas need to be implanted, to facilitate direct node-base station communication, or a multi-hop topology needs to be in place for data delivery. Although most sensor networks studied in the literature have less than 500 nodes, we intend to
study the effectiveness of our scheme in the presence of a large set of sensor nodes, and its impact on the overall success in attack detection. We perform experiments for varying node deployment densities on a 100m x 100m network. The values of the node deployment density, denoted as $N$, are: 128, 256, 512, 1024 and 2048.

The simulation experiments are performed for two types of adversarial nodes, namely, injected nodes and laptop-class nodes. It is assumed that all nodes are equally likely candidates for loss owing to failures, battery exhaustion or compromise. The GN and mGN nodes in the network participate in routine sensing operations, in addition to participation in the attack detection process. Therefore it may be safely presumed that the task of selectively identifying and launching attacks against such nodes by the adversary-class is nontrivial. We also assume that sensor nodes have a single interface for both transmit and receive operations. We consider a standard sensor node with average energy consumptions for transmission and reception as: $E_{\text{trans}} = 100$ nJ/bit and $E_{\text{recv}} = 50$ nJ/bit, respectively (Song, 2005). We also assumed that a typical sensor node has a maximum radio range of 50 meters (Krohn et al., 2006).

The following parameters were incorporated in the simulation setup for the scheme:

- $SR$: The transmission range of a sensor node $\sim 50$m.
- $\Delta_{\text{opt}}$: Time epoch length for detection scheme convergence (Calculated based on Equation 4.8).
- $\alpha$: Application aspect value.
- $TI$: Traffic intensity in terms of packets/second.
• $TI_e$ (Traffic Intensity): Packets generated towards the $r$ target nodes during a given time epoch ($\Delta_{opt}$), in terms of energy resource usage by the target nodes.

• Number of Target nodes: $r = 10\%$ of $N$.

• Number of Detector (GN) nodes: $n$.

If the current epoch of time is defined as $\Delta_i$, then the subsequent epoch of time, based on Lemma 4.1, is given by:

$$
\Delta_{i+1} = \Delta_i + \Delta_{opt}(3.d1 + \frac{n}{m}.d2 + m.d3)
$$

(5.1)

where,

d1 = Average GN to GN communication delay

d2 = Average GN to mGN communication delay

d3 = Average mGN to BS communication delay

The time epoch length is dimensionalised into the unit of time (seconds), and is large enough to accommodate the entire communication phase of the attack detection scheme.

For each value of $N$, we generate simulation plots for varying intensities of traffic generated in the network. These traffic intensities, denoted as $TI$, are inclusive of both normal and attack traffic. For a standard sensor network, the frequency of packet arrivals at a particular node depends on the node’s topological placement in the network. For a network with $N = 1024$, with 10 operational cluster heads, the total number of packets expected by each cluster head during a given time epoch, is approximately equal to 100 sensory packets, for a network with a constant taxonomy i.e. 1 packet generated by
every node per time epoch, for delivery to the cluster head. Therefore, a standard cluster head receiving in excess of 100 packets per time epoch, can be considered to be under attack. We define $TI = 500$ packets/sec, for a scenario with time epoch $= 1$ second, as network traffic with unusual intensity, intended to flood a victim node, and exhaust its limited energy resource. The traffic arrival rate is modeled as a Poisson process with exponential interarrival times. The convergence phase of the scheme is the time required to execute the communication phase of Algorithm 4.1, wherein the GN and the mGN nodes perform pattern reconstruction, by exchanging subpattern values amongst themselves, to confirm an attack.

### 5.2.2 Energy Decay Rates

As described in Section 4.3, the application aspect value, $\alpha$, is a system parameter defining the significance of the accuracy in attack detection over the energy utilisation rates of the GN/mGN nodes or vice versa. The normalised value of $\alpha$ between $\{0.0-1.0\}$, is defined at network initialisation time. The value of the parameter $k$ of the $mSelect$ algorithm is inversely dependant on the selected value of $\alpha$. Large values of $\alpha$ signify the need for achieving higher accuracies in attack detection. Therefore, selection of a large $\alpha$ value (close to unity) will generate smaller values for the parameter $k$, of the $mSelect$ algorithm, effectively leading to the designation of a large number of mGN nodes for the detection scheme. Similarly, smaller values of $\alpha$ will lead to the generation of higher values of $k$, and in essence will lead to fewer numbers of mGN nodes in the network.

In Figure 5.1, we illustrate the overall energy consumption rates of the GN nodes of the network. The energy consumption rates of the GN nodes increase
with corresponding increases in the value of $\alpha$. For instance, for $N=128$ and $\alpha=0.1$, the energy consumption is at 51 $\mu$J/sec, whereas for $\alpha=0.9$, the nodes utilise 84 $\mu$J/sec. This increase in the energy consumption of the GN nodes for higher $\alpha$ values is attributed to the corresponding decrease in the size of the time epoch length $\Delta_{opt}$, for achieving higher accuracies in the attack detection rates. It may be noted here that increasing values of $N$ lead to improved energy consumption rates for individual GN nodes, as the proximity of the GN nodes leads to reduced communication distances that need to be traversed by the GN communication messages.

Figure 5.1: GN Node Energy Utilisation Rate vs. Application Aspect Value ($\alpha$). The peak energy consumption rates in $\mu$J/sec ($\alpha = 0.1$) is 86 for $N=2048$. The energy consumption rate of 17 $\mu$J/sec is lowest for $\alpha=1.0$ and $N=128$.

We illustrate the energy consumption rates of the mGN nodes of the detection scheme in Figure 5.2. For all node deployment densities, the mGN nodes can be seen to consume more energy than the GN nodes. This is due to the additional tasks imposed on the mGN nodes for message reception, analysis, and delivery to the base station, as compared to the standard tasks of a GN node, which involve observation and reporting of traffic flow data to a closely located mGN node. The mGN nodes show a decrease in the

150
energy consumption rate for corresponding increases in the value of $\alpha$. This trend occurs due to the reducing number of mGN nodes selected for the attack detection process, for decreasing values of $\alpha$. Therefore, the energy consumption rate for $N=2048$ and $\alpha=0.1$ is close to $360 \, \mu J/sec$, whereas for $\alpha=0.9$, it is only $126 \, \mu J/sec$. For lower values of $N$, the energy consumption rate of the mGN nodes is lower, as fewer number of GN nodes will collaborate with their designated mGN nodes, and therefore will lead to lesser energy utilisation rates. However, even for lower values of $N$, the overall energy consumption rate of the mGN nodes reduces with increasing $\alpha$.

For low node deployment densities, the energy consumption rates of the mGN nodes are higher, as compared to networks with higher values of $N$. Communications over longer distances that need to be performed in less dense networks lead to higher energy utilisation rates for lower $N$. For $N=128$ and $\alpha=0.1$, the mGN nodes consume $126 \, \mu J/sec$, whereas, for $N=2048$ and $\alpha=0.1$, the mGN nodes consume $360 \, \mu J/sec$. Similarly, for $N=128$ and $\alpha=0.1$, a GN node will consume $21 \, \mu J/sec$, whereas for $N=2048$ and $\alpha=0.1$, the energy consumption rate of a GN node is $84 \, \mu J/sec$. For higher node deployment densities, more numbers of GN nodes communicate with each mGN node. Therefore, the overall energy consumption rate of the mGN nodes depicts an increase for corresponding increase in the value of $N$.

In Figure 5.3, we illustrate the total number of mGN nodes selected by the $m$Select algorithm for varying values of $\alpha$ and $N$. As can be observed, for lower values of $\alpha$, the total number of mGN nodes selected are very low. For instance, for $N=128$ and $\alpha=0.1$, the total number of selected mGN nodes is 2. Smaller values of $\alpha$ imply lesser significance on the accuracy in attack detection, and more significance applied to the energy conservation of the
The peak energy consumption rates in $\mu$J/sec ($\alpha = 0.1$) is 352 for $N=2048$. The energy consumption rate of 32.3 $\mu$J/sec is lowest for $\alpha=1.0$ and $N=128$.

GN and mGN nodes. Therefore, the detection rate accuracy, affected by the longer convergence delays associated with the communication phase of the attack detection process (as each node has a single interface for message transmission and reception), has a corresponding energy conservation factor associated. Similarly, higher values of $\alpha$ imply more significance given to the accuracy in attack detection as compared to the energy conservation of the GN/mGN nodes. In such scenarios, more number of mGN nodes are selected by the mSelect algorithm, so as to reduce the convergence delays of the attack detection scheme, effectively increasing the attack detection rate. However, as illustrated in Figure 5.2, the higher energy consumption rates of the mGN nodes will lead to rapid reduction of their respective lifetimes.

Considering the significantly high energy utilisation rates of mGN nodes as compared to the GN nodes, the presence of a large number of mGN nodes will incur significant overhead on the network, and will lead to reduced lifetimes of a larger number of sensor nodes. Therefore, from an energy consumption
perspective, the fewer the number of mGN nodes, the longer the lifetime of the sensor network. The set of mGN nodes for the detection scheme are selected based on Algorithm 4.2, which operates by reducing any redundancies in node selection.

In Equations 5.2 and 5.3, we define the standard energy decay rates for a GN and an mGN node, per time epoch of length $\Delta_{opt}$. Each GN node receives exactly two traffic observation packets from its neighbouring GN nodes within each time epoch. Therefore, the energy associated with receiving packets, is given by: $2.E_{recv}$. In addition, the GN nodes monitor traffic flow in the network. The total energy usage associated with receiving all packets in a single epoch of time is given by: $2.E_{recv} + pkts(用水).E_{recv}$. Each GN node communicates with exactly three other nodes, namely, two peer GN nodes, and one mGN node, during each time epoch. Considering the average distance between any two GN or mGN nodes to be $d_{GN-mGn}$, the energy usage associated with the transmission of data by a GN node in a single time epoch, is given by: $3.E_{trans}.d_{GN-mGn}$. 

Figure 5.3: Number of mGN Nodes vs. Total Number of Nodes. The peak energy consumption rates in $\mu J/sec$ ($\alpha = 0.1$) is 86 for $N=2048$. The energy consumption rate of $17 \mu J/sec$ is lowest for $\alpha=1.0$ and $N=128$. 

153
The mGN nodes receive packets from \( \frac{n}{m} - 1 \) GN nodes within each time epoch. They are also responsible for transmission of one packet to the base station, located at an average distance of \( d_{mGn-BS} \). The total costs of receiving and transmitting data packets by the mGN nodes are given by: 

\[
[(\frac{n}{m} - 1).E_{recv}] \text{ and } [E_{trans}.d_{mGn-BS}^4],
\]

respectively.

\[
\mu_{gn} = \frac{2.E_{recv} + 3.E_{trans}.d_{GN-mGn}^2 + pkts(obsv).E_{recv}}{\Delta_{opt}}
\]

(5.2)

\[
\mu_{mgn} = \frac{\frac{n}{m} - 1.E_{recv} + E_{trans}.d_{mGn-BS}^4}{\Delta_{opt}}
\]

(5.3)

where,

\( E_{recv} \) = Energy required to receive \( k \) bits = 50 nJ/bit.

\( E_{trans} \) = Energy required to transmit \( k \) bits over \( a = 100 \) nJ/bit.

distance of \( d \) meters.

\( pkts(obsv) \) = Total packets observed by the detector node.

\( \Delta_{opt}(seconds) \) = Time epoch length in seconds = \( t_{i+1} - t_i \).

\( \mu \) = Energy consumption rate.

In Table 5.1, we illustrate the values of \( \Delta_{opt} \) in seconds, for variations in the values of \( \alpha \) and \( TI_e \), for \( N = 1024 \), packet length=1 byte and \( n = 100\% \). This value of \( \alpha \) defines whether more significance is to be given to the conservation of energy of the GN and mGN nodes, or to the conservation of energy of the target nodes (Section 4.3). Higher values of \( \alpha \) selected at network initialisation time define the significance of conserving energy content of the GN and mGN nodes, over the quicker detection of an attack. On the contrary, lower values of \( \alpha \) define the significance of rapid attack detection,
over the need for conservation of the energy content of the GN and mGN nodes.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$TI_e (\mu J/\text{packet})$</th>
<th>$\Delta_{opt} (\text{secs})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>40</td>
<td>4.8</td>
</tr>
<tr>
<td>0.5</td>
<td>40</td>
<td>2.15</td>
</tr>
<tr>
<td>0.95</td>
<td>40</td>
<td>1.61</td>
</tr>
<tr>
<td>0.1</td>
<td>400</td>
<td>2.31</td>
</tr>
<tr>
<td>0.5</td>
<td>400</td>
<td>0.68</td>
</tr>
<tr>
<td>0.95</td>
<td>400</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Table 5.1: $\Delta_{opt}$ (seconds) values for variations in $\alpha$ and $TI_e$

The impact of the variation of the $\Delta_{opt}$ value on the energy resource utilisation of the detector as well as the mGN nodes, for a network with $N=1024$ and $TI = 500$, is illustrated in Figure 5.4. In addition, the figure also illustrates the rate of decay of the energy content of a target node under an attack ($TI=500$). The optimal length of a time epoch is computed based on Equation 4.8, and is affected by the following network and algorithmic parameters: $n$, $N$, $m$, $TI_e$ and $\alpha$, apart from the energy utilisation rates, which are fixed system parameters.

The mGN nodes of the network participate in active reception of a large numbers of packets within each epoch of time, and are responsible for further forwarding of a verdict signal to a base station, over a longer communication channel. These tasks are performed by the mGN nodes, in addition to their detection tasks, as well as routine sensory operations. Therefore, the energy decay rates for mGN nodes are significantly higher as compared to GN nodes. Networks with low node deployment densities will have fewer numbers of nodes, with increased per-node overhead associated with the GN and
mGN tasks. On the contrary, networks with higher node deployment densities will have reduced dependance on a few select nodes, operating as mGN nodes. Therefore, higher node deployment densities will yield lower per-node overhead, associated with the mGN tasks. Higher node deployment densities will also increase the values of $\Delta_{opt}$, thus leading to less frequent convergence of the detection scheme, and lower energy decay rates for the GN and mGN nodes.

Larger values of $\Delta_{opt}$ will cause the detection scheme to converge on a less frequent basis, and therefore will yield lower energy decay rates for the GN nodes. However, the increasing value of $\Delta_{opt}$, attributed to increasing values of $\alpha$, will lead to the selection of fewer mGN nodes in the network, and therefore, the per-mGN node energy consumption rate increases. Smaller values of $\Delta_{opt}$ will lead to more frequent convergence of the scheme, and therefore higher energy consumption rates are observable for the GN nodes.

For larger values of $\Delta_{opt}$, the target nodes under an attack will have a higher percentage of their energy content depleted before an attack against them is actually detected. For instance, for $\Delta_{opt} = 4.8$ seconds ($\alpha=0.1$ and
$TI_e=40$), nearly 970 $\mu$Joules are consumed by the target node each second, as compared to 100 $\mu$Joules consumed, for $\Delta_{opt} = 0.49$ seconds ($\alpha=0.95$ and $TI_e=400$).

Increasing values of $k$ (decreasing $\alpha$), will lead to lesser overlaps in the $k$-lists of each of the GN nodes, exchanged with the base station at network initialisation time. As a result, fewer number of mGN nodes are selected. The reducing number of mGN nodes lead to an increase in the per-mGN node energy utilisation rate. Considering the high energy utilisation costs associated with the mGN nodes, it is recommended to have as few mGN nodes operating in the network as possible. However, the accuracy in attack detection of the scheme will in effect diminish, as will be elaborated in the following subsection.

### 5.2.3 Attack Detection Rates

The attack detection rate is defined as the ratio of the total number of attack packets classified correctly, over the total number of attack packets, given by:

$$\text{Attack Detection Rate} = \frac{\text{Total Observed Attack Packets}}{\text{Total Attack Packets}}$$ (5.4)

In Figure 5.5, we analyse the effect of variation of the value of the application aspect value, $\alpha$, on the attack detection rate. As seen from the figure, the attack detection rate is higher for $\alpha$ close to unity. For instance, for $\alpha = 1.0$, the attack detection rate is 38% for $N=128$, whereas, for $\alpha = 0.1$, the detection rate is only 10%. For $N=2048$, the detection rate is 92% for $\alpha=0.1$, and is 86% for $\alpha=1.0$. A similar increase in the attack detection rate for increasing values of $\alpha$ is observable for the other node deployment densities.
Figure 5.5: Attack Detection Rate vs. Application Aspect Ratio ($\alpha$) for $TI = 500$. The peak detection rate ($\alpha = 1.0$) is 38% for $N=128$, 65% for $N=256$, 71% for $N=512$, 84% for $N=1024$ and 92% for $N=2048$. The detection rate is lowest for $\alpha=0.1$: 10% for $N=128$, 31% for $N=256$, 47% for $N=512$, 61% for $N=1024$ and 86% for $N=2048$.

For purposes of our simulation experiments to compute the attack detection rate, false alarm rates and the time epoch length, we have considered the value of $\alpha$ to be 0.95, to study the peak accuracies in attack detection. In Figure 5.6, we illustrate the attack detection rate for a network with node deployment density, $N = 128$. The intensity of the total traffic in the network, inclusive of attack as well as normal packets, is varied from 50 packets/sec to 500 packets/sec. The total number of detector nodes ($n$) is also varied from 1% to 100%. For $TI = 50$, the detection rate reaches nearly 72%, when the number of detector nodes = 100%. Due to the low node deployment density of this network, the detector nodes in the network cannot reconstruct, in their entirety, accurate traffic observation patterns, as packets penetrating the network from unobserved regions of the network are not accounted for, by the detector nodes. Therefore, the detection rate does not cross 72%, even with 100% GN nodes in the network, and low traffic intensities.
For higher values of $TI$, the detection rates further degrade, with the detection rate being only 30%, for $TI = 500$ and $n = 100\%$. This is because high $TI$ values imply larger numbers of packets penetrating the network, whilst the attack detection scheme is still in the process of convergence. These packets remain unobserved by the detector nodes, and thus the performance of the scheme in terms of the detection rates, degrades with increasing traffic intensities.

![Figure 5.6: Attack Detection Rate vs. Detector Node Ratio for $N = 128$. The peak detection rate is approximately 72% for low traffic intensity and $n = 100\%$. For $n < 10\%$, the detection rate is negligible for all traffic intensities.](image)

For the $N=256$ scenario (Figure 5.7), smaller values of $T.I.$ require fewer numbers of active detector nodes in the network for reaching higher attack detection rates. For $T.I.=50$, with roughly 35% detector nodes, the detection rate is nearly 53%, as compared to the $N=128$ scenario, where the detection rate was less than 33%. The increase in the densities of nodes deployed in the network improves the chances of detector node presence in all regions of the network. As a consequence, higher detection rates are witnessed for fewer number of operating detector nodes. For larger values of $TI$, the scheme shows good improvements over the $N=128$ scenario, with the detection rate
approaching nearly 70% for $T_I=500$, as compared to $N=128$, where the detection rate did not exceed 33%, for the same traffic intensity. Larger numbers of detector nodes in the network facilitate the verification and reconstruction of a pattern depicting observed network traffic, with higher degree of accuracy. Therefore, higher detection rates are observed.

In the $N=512$ (Figure 5.8) scenario, attack detection rates peaked to nearly 90% in the presence of as few as 20% detector nodes in the network. The higher density of node deployment in these networks, assure that fewer detector nodes are required to achieve higher success in the attack detection process. This is because the higher numbers of detector nodes in the network help accurately reconstruct traffic observation patterns, from individual readings of the large number of attack detector nodes, thus leading to higher detection rates. The parallel nature of execution of the communication phase of the detection scheme (Algorithm 4.1), wherein the GN and mGN nodes coordinate to reconstruct the complete pattern of observed traffic, reduces the overhead associated with having higher number of detector nodes on the convergence delay of the detection scheme.

Again, higher values of $T_I$ yield lower attack detection rates, when fewer numbers of detector nodes are operational in the network, for this network scenario, with $T_I=500$ yielding a detection rate of 82%, with $n = 100\%$. This is because of the larger numbers of attack packets penetrating the network, and remaining undetected, during the convergence of the communication phase of the detection scheme.

In the $N=1024$ scenario (Figure 5.9), lower values of $T_I$ require fewer number of active detector nodes in the network for reaching relatively high attack detection rates. For $T_I=50$, with $n=22\%$, the detection rate is nearly
Figure 5.7: Attack Detection Rate vs. Detector Node Ratio for $N = 256$. The detection rate approaching 70% even with high traffic intensities ($TI=500$), and fewer than 100% $n$ nodes required to attain high detection rates.

Figure 5.8: Attack Detection Rate vs. Detector Node Ratio for $N = 512$. Peak detection rate of nearly 90% for as few as 20% detector nodes in low traffic intensities.
93%. The increasing densities of node deployment improve the chances of detector node presence in all regions of the network. As a consequence, very high detection rates are witnessed for fewer number of operating detector nodes. The expected performance improvements owing to the participation of a larger set of detector nodes in the detection process, is subdued for higher values of $TI$. For $TI = 500$, the detection rate is nearly 82%, with 100% detector nodes, whereas for $TI=50$, the detection rate is as high as 96%. However, for larger values of $TI$, the scheme shows reasonable improvements over the previous network scenarios, with the detection rate crossing the 80% mark for $n = 100\%$ for $N=1024$, as compared to all the previous network scenarios, $N=128$, 256 and 512.

![Graph of Attack Detection Rate vs. Detector Node Ratio for $N = 1024$. Peak detection rate of 93% for low traffic intensities. Even high values of $TI$ yield a detection rate of above 80% for higher $n$.](image)

The detection rates for the $N=2048$ scenario (Figure 5.10) are the best amongst all networks. This is because of the ability of the detection scheme to accurately reconstruct traffic observation patterns, even in the presence of high traffic intensities. We can observe a detection rate of nearly 97% for
as few as 50% detector nodes, and $TI=50$, and approaches nearly 90% for $TI=500$.

In Figure 5.11, we illustrate the effect of the variation of the detector node ratio on the attack detection rate for various node deployment densities. For lower values of $n$, very high density networks can sustain a reasonable detection rate. As can be observed, $n=0.05N$ yields a detection rate of only 40% with $N=2048$. Increasing values of $n$ yield higher detection rates for all node deployment densities, with $n=0.75N$ performing nearly as good as the $n=N$ scenario, for $N=2048$. It may be conjectured that the need for having all nodes operating as detector nodes is not essential, if the node deployment density of the network is high. However, the detection rate improvements are reasonably higher for less dense networks, with higher values of $n$.

![Graph](image-url)

Figure 5.10: Attack Detection Rate vs. Detector Node Ratio for $N = 2048$. Peak rate of 97% for low traffic intensities. Only 10-15% of detector nodes needed to achieve high detection rates.
Figure 5.11: Attack Detection Rate vs. Network Size ($N$), for $TI=500$. Higher values of $n$ yield higher detection rates. Larger node deployment densities essential if fewer detector nodes are to be selected, to sustain high attack detection rates.

Summary

The attack detection rates are higher for large node deployment densities, as compared to the low density counterparts. This is because the attack detection scheme relies on both the individual traffic observations by the detector nodes, as well as the subsequent verification and reconstruction of observed traffic subpatterns by the detector nodes as well as the mGN nodes, for classification purposes. The fewer numbers of detector nodes in the network will lead to the reconstruction of less accurate patterns, to depict actual network traffic flow. This phenomenon occurs because of the inability of the scheme to perform attack detection in the presence of unobserved regions of the network under a distributed denial of service attack. A centralised attack scenario will yield comparable detection rates for both low as well as high density networks. We may therefore conclude that higher densities of node deployment yield higher attack detection rates, for fewer numbers of operational detector nodes. We may also infer that higher values of $TI$ will
lead to lower detection rates, due to the larger numbers of attack packets penetrating the network, unnoticed, whilst the scheme communication phase is still converging. The parallelism in the execution of the GN pattern reconstruction process (Communication phase of Algorithm 4.1), helps improve detection rates with corresponding increases in the value of $n$, thus proving the scalability of the detection scheme for denser networks.

**Corollary 5.1.** Higher node deployment densities are required to achieve higher attack detection rates in the presence of large volumes of network traffic.

**Corollary 5.2.** Higher values of $n$ ensure significantly improved detection rates for less dense networks.

### 5.2.4 Pattern Update Rate

The rate of update of pattern values for each of the $r$ target nodes within the $n$ detector nodes affects the detection rate of the scheme. We described the expected pattern update rates in the initialisation sub-phase of Algorithm 4.1 (Distributed attack detection). Following are standard energy consumption models defining the rate of decline of energy within the target sensor nodes (Baig et al., 2006):

- **Flat topology energy consumption model:**

  $$
  \mu_f = \frac{\text{pkts}(\text{recv}).E_{\text{recv}} + \text{pkts}(\text{trans}).E_{\text{trans}}.d^2}{t}
  \tag{5.5}
  $$

- **Cluster-based topology energy consumption model:**

  $$
  \mu_{ch} = \frac{(2n_c + 1).\text{pkts}(\text{recv}).E_{\text{recv}} + \text{pkts}(\text{trans}).E_{\text{trans}}.d^2(n_c + 1)}{t}
  \tag{5.6}
  $$

165
Data Aggregation topology energy consumption model:

\[ \mu_{da} = \frac{(2n_r + 1).pkts(recv).E_{recv} + pkts(trans).E_{trans}d^2(n_r + 1)}{t} \] (5.7)

The pattern update equations defined above can be used to predict a known rate of decline of energy resources of a target node, based on its functionality in the network. For instance, a cluster-head target node can be expected to have its energy resource use dictated by Equation 5.6. However, the actual traffic flow constituting an attack in the network, upon being processed by the target nodes, will require a different pattern update rate value, to be generated, for storage and processing by the detector nodes.

The error rate is defined as an estimate on the inaccuracy of the scheme in detecting attack packets, due to the infrequent update of the subpattern values by the detector nodes. Effectively, this error rate implies that traffic intensity classified as normal during a given time interval, \( \Delta_i \), will be classified as high or anomalous, during a time interval \( \Delta_j, s.t. j > i \). Therefore, the error in detection would culminate into more numbers of false negatives generated by the scheme. In Table 5.2, we illustrate the error rates in detection, for varying pattern update rates for the three network topologies undergoing an attack. Following assumptions have been made for calculating the error rate:

- Maximum number of nodes in a cluster = 10.
- Maximum number of incoming data channels to a aggregation node=3.
- Expected pattern update frequency = 1/1000nJ of energy utilised.
- \( TI = 500 \).
The resulting error rate is the maximum error that can be expected in the detection rates generated in the previous section, if the frequency of the pattern update operation within the detector nodes is not at par, for the above-given parameter values.

```
<table>
<thead>
<tr>
<th>Pattern Updates Per Second</th>
<th>Flat Topology (%)</th>
<th>Cluster-based Topology (%)</th>
<th>Data Aggregation Topology (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>1</td>
<td>0.93</td>
<td>0.87</td>
<td>0.78</td>
</tr>
<tr>
<td>2</td>
<td>0.81</td>
<td>0.65</td>
<td>0.59</td>
</tr>
<tr>
<td>3</td>
<td>0.76</td>
<td>0.50</td>
<td>0.35</td>
</tr>
<tr>
<td>4</td>
<td>0.5</td>
<td>0.41</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.4</td>
<td>0.32</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>0.37</td>
<td>0.26</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>0.27</td>
<td>0.22</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>0.15</td>
<td>0.15</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>0.07</td>
<td>0.07</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
```

Table 5.2: Error rates (%) in detection for varying pattern update frequencies.

The flat topology requires its GN nodes to update their respective threshold values 10 times per second, to achieve 100% accuracy in pattern recognition. In the single-hop model of a flat topology, target nodes need to directly communicate with the base station over longer distances, implying higher rates of energy exhaustion in the target nodes, and therefore the higher frequency of threshold updates.

In a cluster-based topology, the cluster heads are considered as target nodes for the attacks. Shorter inter-hop distances between the individual cluster nodes and the cluster heads imply lesser consumption of the cluster head energy resources. However, the intensive computation and communication requirements imposed on the cluster heads for interaction with both
the cluster nodes as well as the base station compromise the energy gains achieved through shorter inter-hop distances. In this topology, 100% accuracy in attack detection may be achieved by having approximately 10 subpattern update operations taking place every second.

The DA-based topology assumes that all nodes on the path from the source to the sink of a given source-sink path of the network are potential targets. Therefore, the inter-hop distances between adjacent nodes on source-sink paths is reduced significantly. Lesser energy consumption owing to shorter inter-hop distances implies a lower energy decline rate in the target nodes, and therefore, lower frequency of threshold update operations i.e. 4 per second.

5.2.5 False Alarm Rates

The false alarms generated by the detection scheme are a combination of both false positives as well as false negatives. In this subsection, we analyse the false alarm rates of the scheme.

False Positive Rates

The false positive rate is defined as the ratio of the total number of legitimate packets classified by the detection scheme as attack packets, over the total number of packets.

\[
False \ Positive \ Rate = \frac{\text{Packets Incorrectly Labelled Malicious}}{\text{Total Number of Packets}} \tag{5.8}
\]

The false positive rate increases with decreasing numbers of observed sub-patterns, needed for reconstruction of a complete pattern of observed traffic
flow in the network. As elaborated in the detection scheme in Chapter 4, the detection scheme requires exactly half of the total number of observer nodes to communicate with their respective mGN nodes, during each interval of time. On certain occasions, several detector nodes will confirm an attack, based on incorrect peer readings, from both neighbours of the detector nodes. In such scenarios, a large number of incorrect observations will reach the mGN nodes, and a false alarm will be raised, incorrectly indicating an attack in progress.

In Figure 5.12, we illustrate the false positive rate of the detection scheme, for variations in the traffic intensities, $TI$, with $n = N$. The false positive rates are lower for high density networks, due to the high confidence in traffic observation, attained as a result of having several overlapping regions of observation in the network, for reconstruction of a complete traffic observation pattern, by the larger number of detector nodes. Therefore, for the $N = 2048$ scenario, the false positive rate is only 0.5% for $TI=500$, and almost negligible for lower traffic intensities. On the contrary, the false positive rate for a $N=128$ network is more than 5%, for $TI=500$. In such networks, the inability of the detector nodes to cover all regions of the network, reduces the overall detection rate. In turn, the total number of incomplete patterns reconstructed by the detector nodes is higher. Therefore, a large number of false alarms are generated by the detector nodes.

The false positive rates increase with increasing network traffic intensities for all node deployment densities. This is because with higher inflows of traffic, the chances of certain detector nodes neglecting the attack packets whilst the scheme’s communication phase is converging, are higher. If these detector nodes are the ones scheduled to communicate with their respective mGN nodes in the current time epoch, the overall false alarm rates in the
scheme increase. Therefore, higher values of $TI$ will lead to increased false positive rates.

**False Negative Rates**

The false negative rate is defined as the ratio of the total number of attack packets classified as legitimate packets by the scheme, over the total number of observed packets.

\[
\text{False Negative Rate} = \frac{\text{Packets Incorrectly Labelled Legitimate}}{\text{Total Number of Packets}}
\]  

The false negative rate depends on the property of the detection scheme, which demands regular convergence of the scheme at the end of each time epoch. All attack packets penetrating the network at time of scheme convergence remain unnoticed by the detector nodes, and are tagged as false negatives. As seen from Figure 5.13, the false negative rate is higher for networks with low node deployment densities. The reason for such large false negative rates is the same as the one given for the false positive rates i.e.
in larger networks, the absence of detector nodes in certain regions of the network, increases the likelihood of not observing attack packets.

The false negative rates are lower for networks with high node deployment densities. This is because of the presence of multiple detector nodes, with overlapping regions of observation. Therefore, the observed traffic sub-patterns are ascertained with higher accuracies in such networks.

As observed from the figure, for \( N=128 \), the false negative rate is nearly 40\%, for \( TI=500 \), whereas for \( N=2048 \), it is less than 5\%. Networks with higher densities of node deployment have peer readings confirmed by nodes, with overlapping observation regions, thus increasing the accuracy of the reconstructed traffic observation pattern. As a consequence, such networks have lower false negative rates.

For higher \( TI \) values, more numbers of packets traverse through the network towards the target node set, and remain unnoticed, and therefore the false negative rate increases with increasing intensities of network traffic.

![Figure 5.13: False Negative Rate vs. Node Deployment Density (N) for varying Traffic Intensities](image)

Figure 5.13: False Negative Rate vs. Node Deployment Density (\( N \)) for varying Traffic Intensities
Summary

Both the false positive and the false negative rates are higher for networks with low node deployment densities. These rates reflect on the need for having a complete coverage of an entire network, so as to facilitate the reconstruction of accurate patterns of observed network traffic flow, by the GN and the mGN nodes. Networks with high node deployment densities exhibit low false alarm rates. The presence of large numbers of detector nodes in such networks, helps achieve higher accuracies in pattern reconstruction, in turn reducing the false alarms in the network. On the contrary, networks with low node deployment densities have the highest false alarm rates, due to the absence of detector nodes for observation of attack packets, in various regions of the network. It may therefore be concluded that the need for higher accuracies i.e. lower false alarm rates in attack detection, demand the presence of larger numbers of detector nodes in the network.

5.3 Self-Organising Map-based Attack Detection

Neural networks are known to be a very powerful tool in detecting anomalous network traffic in high-performance networks. One such class of neural networks that has been used extensively for intrusion detection and attack detection is the Self-Organising Map (SOM). A SOM is a nonlinear, ordered, smooth mapping of high dimensional input data manifolds onto the elements of a regular, low-dimensional array (Ramadas, 2003). From an attack detection perspective, the resulting geometric map of neurons depicts patterns of
actual network traffic flow. In wireless sensor networks, Self-organising maps have been introduced for generation of optimal data-aggregation trees (Lee and Chung, 2005), and context classification (Catterall et al., 2002).

Self-organising maps are known to exhibit several key characteristics, that make these neural networks ideal for intrusion and denial of service attack detection. These characteristics are: efficient updates of neuron weights, and the ability to express multi-dimensional input patterns as topological relationships on a two-dimensional map (Lichodzijewski et al., 2002). SOMs have been extensively used for intrusion detection. In (Labib and Vemuri, 2003), a SOM-based anomaly detection scheme is proposed, for classifying network traffic, with the intent of detecting denial of service attacks. In the proposed scheme, the SOM is trained with normal network traffic data, and subsequently all real-time data is clustered into a winning neuron, not labeled during SOM training, is classified as an attack. In (Mitrokotsa and Douligeris, 2005), SOMs have been used for detecting denial of service attacks. The authors propose the use of emergent properties i.e. additional neurons at higher layers, to perform clustering of the observed network traffic, for detection purposes.

In the absence of a decentralised mechanism for detection of distributed denial of service attacks in wireless sensor networks, individual traffic observations by sensor nodes, need to be transmitted to the base station for further analysis. It may be noted that we are considering a scenario wherein the GN pattern learning mechanism is non-existent. Therefore, all processing and analysis of the data needs to be done at the base station. Considering the centralised availability of all data associated with observed traffic in
the form of patterns, a neural network-based approach for clustering of traffic observation patterns into an appropriate cluster, can be applied in these scenarios. We implement a SOM-based centralised attack traffic clustering mechanism, to test the effectiveness of such an approach. We benchmark the simulation results for our proposed distributed attack detection scheme, against corresponding results obtained from a centralised SOM-based DDoS attack pattern clustering scheme, operating on the base station of the wireless sensor network.

The SOM application is responsible for clustering of network traffic packets into one of \( l \) neurons of the lattice map, based on the proximity of the neuron weight to the weight of the input vector, in terms of Euclidean distances. The decision layer does the actual classification of network traffic into attack or normal, based on the inputs received from the SOM layer.

![SOM overlay on base station](image-url)

Figure 5.14: SOM overlay on base station
The SOM-based attack detection scheme has two phases of operation, namely, learning and classification.

5.3.1 Learning Phase

During this phase, aka training phase, the SOM application is introduced with a set of learning patterns, to train the $l$ map neurons to map data points of the input vector, onto the array of neurons. The mapping process is competitive. It is performed by introducing data points of the input vector to each of the $l$ neurons of the map, one vector at a time. For each input vector, the neuron with the closest weight in distance (Euclidean or Hamiltonian), to the input data point, is declared the winner. Subsequently, the weight of the winner neuron is adjusted to ensure that its values are inclined more towards data points similar in characteristics to the current input data point. In addition, for each of the input vectors, neighbours of the winning neuron have their weights updated as well. A neighbourhood function needs to be defined to calculate the neighbours of a given winner neuron. Typically, the neighbourhood function is taken as either Gaussian or Bubble. The $k$ dimensional values within the neighbouring neurons of the winner are adjusted accordingly.

At the end of training phase, each neuron of the map is labelled as an attack or normal class, based on a majority count of the classes of input patterns, for which the neuron was declared the winner. The labelling function is defined as:
\[ label_l = \begin{cases} 
\text{attack}, & t^l_{\text{attack}} > \text{thresh}_{\text{attack}} \\
\text{normal}, & t^l_{\text{normal}} > \text{thresh}_{\text{normal}} 
\end{cases} \]

where, \( t^l_{\text{attack}} \) is the total number of attack packets for which neuron \( l \) was declared the winner, and \( t^l_{\text{normal}} \) is the total number of normal packets for which neuron \( l \) was declared the winner. The \( \text{thresh}_{\text{attack}} \) is the threshold of attack packets, if observed by a neuron \( i \), will lead to it being labelled an attack class. Similarly, \( \text{thresh}_{\text{normal}} \) is the threshold of normal packets, if observed by a neuron \( i \), will lead to it being labelled a normal class, where \( \text{thresh}_{\text{normal}} = 1 - \text{thresh}_{\text{attack}} \).

### 5.3.2 Data Classification

During the classification phase of the scheme, the \( k \)-dimensional weight arrays associated with the input vectors are compared with the weight vectors of the \( l \) neurons of the map. The neuron with the closest match is declared a winner, and the corresponding input vector is classified accordingly. The decision making layer of the scheme generates the final verdict on the classification of the observed input pattern vector into attack or normal traffic flow.

### 5.3.3 Parameter selection

The training phase of the SOM algorithm is performed offline on the base station, using the patterns generated as part of the sample data. Prior to execution of the training phase, the SOM application is initialised with the selected SOM training parameter values. The initial values selected for the map are crucial in defining a good quality map layout at the end of the training.
phase. The weights must be within the range of values of the \( r \) dimensional pattern vectors in the sample data set. Using simulations, we generated parameter values for the initial map dimensions, based on the sample data consisting of both attack and normal network traffic. The map dimensions are selected such that the ratio of the map dimensions is proportional to the square root of the calculated ratio.

The value of \( \Delta_{opt} \) is selected based on an \( \alpha = 0.95 \), and other values are varied based on the node deployment density \( (N) \). The optimal map size is a function of the size of the training data set, and the \( k \)-dimensional values of the training data. A 100% detector node ratio \( (n) \) is considered for all simulations. A total of 5000 traffic packets comprising of both attack as well as normal packets are introduced to the SOM application during the learning phase. Subsequently, another 5000 packets are introduced to the SOM application for actual classification. The following parameter values are selected based on simulations:

- \( m : 20 \)
- \( p : 18 \)
- \( \sigma : 27 \)
- \( \alpha' : 0.5 \)
- \( thresh_{\text{attack}} : 0.5 \)

where the map is of size \( m \times p \), Gaussian radius is given by \( \sigma \) and the learning rate parameter is given by \( \alpha' \).
5.3.4 Evaluation

We performed simulations to generate results for the attack detection rates, false positive rates, and the false negative rates for varying values of $N$, and varying network traffic intensities.

Detection Rates

In Figure 5.15, the attack detection rate during initial lifetime of the network (post-initialisation), is plotted for varying node deployment densities. The detection rate is nearly 92% for high node deployment densities and low traffic intensities, whereas for $N = 128$, the detection rate is only 65%, even for low traffic intensities. The lower node deployment density networks have lesser detection rates as compared to the higher density networks. The presence of fewer detector nodes in the low density network scenarios lead to fewer successes in the attack detection process, due to the incompleteness in the pattern vectors generated for transfer to the base station, for subsequent clustering by the SOM application.

![Graph showing initial attack detection rate vs. network types for varying traffic intensities.](image)

Figure 5.15: Initial Attack Detection Rate vs. Network Types for varying traffic intensities. A peak value of 92% is achieved for $N=2048$ and $TI=50$. The lowest detection rate is for $N=128$ and $TI=500$. 

178
Higher traffic intensities imply more numbers of packets penetrating the network, unnoticed, during the convergence of the detection scheme, and therefore larger values of $TI$ lead to lower detection rates for all values of $TI$.

In Figure 5.16, we plot the average detection rate of the SOM-scheme versus the rate of decline of energy content within the target nodes, for $TI=500$. The rate of decline is defined as the percentage reduction in the energy content, as compared to the initial energy values of the target nodes. It may be noted that the rate of decline of energy content in the target nodes depends on the traffic intensity as well as the node’s topological commitments. Timely detection of distributed denial of service attacks will facilitate resource reallocation by the base station, and hence avoid disruptions in the operations of the network.

![Figure 5.16: Average Attack Detection Rate vs. Rate of Decline of Energy Content in the Target Nodes.](image)

As can be observed, the average detection rate drops significantly with the corresponding decay in the energy content of target nodes. This occurs due to the inability of the SOM application to update pattern values depicting total number of receivable requests by the target nodes, based on the nodes’
energy decline rate. The continuous decay of energy content within the target nodes demands a corresponding update of pattern values, that need to be observed. Considering the inability of the SOM-based approach to update pattern vector values in real-time, the detection rates fall considerably with the passage of time. At the 0.9T mark, roughly 90% of the detector nodes die, leaving incomplete pattern vectors for analysis until the 1.0T mark, where nearly all detector nodes die.

**False Positive Rates**

The false positives for the detection scheme are the total number of normal packets clustered by the detection scheme as attack packets. Considering the lack of detector nodes in certain regions of the network, for smaller values of $N$, incomplete pattern vectors are generated for clustering at the base station. Therefore, the false positive rates are higher for such networks. Figure 5.17 illustrates the false positive rate for the attack detection scheme.

The false positive rate increases with increasing traffic intensities. This is because the pattern vectors generated for traffic analysis are more accurate, when fewer numbers of packets penetrate the network at time of scheme convergence. As a consequence the accuracy of the classification performed at the base station is higher for low $TI$ values. Therefore, fewer false positives are observed for lower intensities of traffic flow.

From Figure 5.18, we can observe that the average false positive rate increases with the increasing rates of decline of target node energy content. As with the false negative rates, the reducing energy contents of the target nodes require corresponding updates in the pattern values, unaccomplished by the Self Organising Map application. Therefore, the false positive rates
Figure 5.17: Initial False Positive Rate vs. Network Types for varying traffic intensities. A high false positive rate of nearly 14% is observed for \( N=128 \) and \( TI=500 \), whereas a very low false positive rate of approximately 2% is observed for \( N=2048 \) and \( TI=50 \).

show a significant and steady increase with the decline of node resources over time.

**False Negative Rates**

The false negatives for the scheme are defined as the total number of attack packets classified by the detection scheme as legitimate traffic packets. The false negatives for the detection scheme are a summation of the total number of attack packets clustered by the SOM application into a normal cluster, and the total number of attack packets that remain undetected at time of application convergence (i.e. node-to-base station communication). The comparison of the initial false negative rates for varying network types is illustrated in Figure 5.19. The false negative rates are lower (< 10%) for larger values of \( N \), as compared to corresponding false negative rates from smaller values of \( N \). The absence of detector nodes leads to generation of incomplete pattern
Figure 5.18: Average False Positive Rate vs. Rate of Decline of Energy Content in the Target Nodes. $TI=500$ A peak false positive rate of 30% is observable for all $N$ values, when 10% of the target node’s energy content is depleted.

vectors for subsequent analysis by the SOM application, and therefore causes an increase in the false negative rates of the scheme.

The false negative rate increases with increasing values of $TI$, as higher traffic intensities also lead to higher numbers of attack packets entering the network within the same convergence time period, and therefore the total number of attack packets that remain unobserved during the same time epoch length is higher. Therefore, higher traffic intensities will lead to higher rates of false negatives.

In Figure 5.20, the false negative rate for progressing lifetimes of the detector nodes is given:

As can be observed, the average false negative rate increases with corresponding decrease in the lifetime of the target nodes. A peak false negative rate of 62% is observable when nearly 12% of a target node’s lifetime is depleted. It may be observed that after a certain lifetime of a target node is
Figure 5.19: Initial False Negative Rate vs. Node Deployment Density ($N$) for varying traffic intensities. The highest false negative rate value observed is 30% for $N=128$ and $TI=500$, and the lowest value observed is 5% for $N=2048$ and $TI=50$.

Figure 5.20: Average False Negative Rate vs. Rate of Decline of Energy Content in the Target Nodes. $TI=500$ A peak false negative rate of 62% is observable for all $N$ values, when 13% of the target node’s energy content is depleted.
reached, 12% in this case, the role of additional attack detector nodes, to re-
constitute a complete pattern vector for analysis at the base station, becomes
ineffective. Therefore, even higher values of $N$ do not affect the false alarm
rates of the scheme.

The observations made in this section entail towards the need for having
pattern update mechanism in place to ensure maintenance of updated pattern
values for the target nodes of the network, so as to achieve reasonable success
in attack detection.

**Energy Decay Rate**

In this section, we analyse the energy decay rates of the attack detector
nodes of the network. As elaborated earlier, attack detector nodes observe
and generate attack pattern vectors at the end of each time epoch $\Delta$, for
communication to the base station. The total energy decay rate of the detec-
tor nodes is therefore a function of the communication cost, $\text{Cost}_{\text{comm}}$, and
the computation cost, $\text{Cost}_{\text{comp}}$. With the $\text{Cost}_{\text{comp}}$ associated with storage
and generation of pattern vectors within a detector node being negligible (see
Chapter 4), the energy decay rate is approximately equivalent to $f(\text{Cost}_{\text{comm}})$.
The $\text{Cost}_{\text{comm}}$ in turn is a function of the network dimensions and the node
deployment densities.

The average rate of decline of energy resources, $L_n$, of a detector node $n$
in the self organising map-scheme is given by:

$$\mu_{\text{som}} = \frac{\text{pkts}(\text{recv}).E_{\text{recv}} + \text{pkts}(\text{trans}).E_{\text{trans}}d^4}{t}$$  \hspace{1cm} (5.10)
As can be seen from Table 5.3, energy exhaustion rate is higher for networks with lower densities of node deployment.

<table>
<thead>
<tr>
<th>Node Deployment Density (N)</th>
<th>Energy Decay Rate (µJ/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>346</td>
</tr>
<tr>
<td>256</td>
<td>173</td>
</tr>
<tr>
<td>512</td>
<td>136</td>
</tr>
<tr>
<td>1024</td>
<td>122</td>
</tr>
<tr>
<td>2048</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 5.3: Energy Decay Rates for the SOM-based centralised detection scheme.

This is due to the longer average distances between the detector nodes and the base station in such scenarios. On the contrary, networks with higher node deployment densities have lesser average distances for coverage by the pattern messages from the detector nodes to the base station, and therefore, smaller energy decay rates.

**Summary**

The attack detection rates of the SOM-based scheme start reasonably high at initialisation time, Figure 5.15, for all node deployment densities. For instance, the $N=2048$ network exhibits a detection rate greater than 85% for all traffic intensities. However, the scheme fails to sustain the high detection rates for long, as can be inferred from Figure 5.16. This phenomenon occurs due to the inability of the SOM to update its trained neurons, to reflect energy decay rates of the target nodes. Therefore, the detection scheme is totally ineffective when 90% of a detector node’s lifetime is reached. We can therefore conclude that a neural network-based approach is not very effective
in detection of distributed denial of service attack patterns in wireless sensor networks.

The false alarm rates show a similar trend to the detection rate, albeit inversely. The false positive rate of the scheme for \( N = 2048 \), Figure 5.18 shows a steady increase from nearly 5\% at network initialisation time, to nearly 27\% when 10\% of the target node’s energy content is depleted. Similarly, the false negative rates also approach close to 60\%, around this time. The inability of the SOM-based scheme to update pattern values, and re-train the neurons, to achieve higher accuracies, are clearly exhibited in these figures. We may therefore infer that the need for a constant pattern update mechanism in place, accompanied with distributed pattern recognition, are essential to achieve higher rates of success in attack detection.

### 5.4 Comparative Analysis

In Table 5.4, the average detection rates for our proposed distributed attack detection scheme (Chapter 4), are compared with corresponding detection rates, both initial (immediately after network initialisation), and average over a target node’s lifetime, of the SOM-based attack detection approach, for the following parameter values:

- \( \alpha = 0.95 \).
- \( TI = 500 \).
- \( n = 100\% \).

The distributed attack detection scheme consistently yields high attack detection rates, as compared to the SOM-based approach. For \( N = 128 \), both
the distributed attack detection as well as the SOM-based approach yield an average detection rate of 56%. However, the SOM-based scheme has consistent degradation in its performance over the period of the lifetime of the target nodes. The average detection rate of the SOM-based approach is only 9% before the energy content of the target node is completely depleted. For all values of $N$, the SOM-based scheme has lesser success in attack detection, both in terms of the initial detection rates, as well as the average detection rates. The reason for this consistent degradation in detection rates of the SOM-based approach is the inability of this technique to perform retraining of the SOM neurons whilst the attack detection is taking place, after the neurons are initially trained, at the base station. The delays associated with SOM re-training at runtime, hinder the possibility of having such an approach applied in such a network environment, for attack detection. Comparisons between the average detection rates for the two schemes signify the need for having distributed pattern recognition in place, to sustain high attack detection rates over the entire lifetime of the network.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Distributed Scheme</th>
<th>SOM-based Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial</td>
<td>Average</td>
</tr>
<tr>
<td>128</td>
<td>56</td>
<td>9</td>
</tr>
<tr>
<td>256</td>
<td>72</td>
<td>10</td>
</tr>
<tr>
<td>512</td>
<td>76</td>
<td>11.7</td>
</tr>
<tr>
<td>1024</td>
<td>87</td>
<td>13.4</td>
</tr>
<tr>
<td>2048</td>
<td>94</td>
<td>13.7</td>
</tr>
</tbody>
</table>

Table 5.4: Detection Rate Comparison - distributed detection and SOM-based schemes
In Table 5.5, we compare the initial and average false alarm rates of the SOM-based detection scheme, with the average false alarm rates of the distributed detection scheme. For all values of $N$, the false alarm rates (both false positive rate and false negative rate) are higher for the SOM-based scheme. The false positives of the two schemes are lower as compared to the false negative rates. In both the detection schemes, detector nodes communicate with their respective decision-making nodes, i.e. GN nodes with their designated mGN nodes, and detector nodes with the base station in the SOM-based scheme. During this communication phase of the schemes, the total numbers of malicious packets penetrating the network, and remaining unnoticed, increment the false negatives. On the contrary, the false positive rates are primarily influenced by the accuracy of the algorithm utilised in the detection scheme. For the distributed scheme, the false positives are generated when GN nodes generate attack signals, based on incorrect peer readings, for delivery to their respective mGN nodes. In the SOM-based scheme, the false positives are generated based on the incorrect clustering of attack packets in the neurons, labeled as normal, during the initialisation and training phases.

| $N$  | Distributed Scheme |  | SOM-based Scheme |  |  |
|------|--------------------|  | Initial | Average |  |  |
|      | FP Rate | FN Rate | FP Rate | FN Rate | FP Rate | FN Rate |
| 128  | 5.2      | 39      | 14.5    | 29      | 30      | 60.2 |
| 256  | 3.4      | 25      | 23      | 11.4    | 29.6    | 59.3 |
| 512  | 2.9      | 22      | 16.7    | 8.4     | 29.1    | 58.3 |
| 1024 | 1.6      | 10      | 9.3     | 4.7     | 28.8    | 57   |
| 2048 | 0.4      | 4       | 8.1     | 4.1     | 28.5    | 56.9 |

Table 5.5: False Alarm Rate Comparison - Distributed detection scheme and SOM-based schemes
The initial false negative rates are more comparable for both the schemes. This is due to the property of both schemes which demands frequent communications both at the inter-node level, as well as at the node-base station level. The centralised approach of the SOM-based approach yields better initial false negative rates than the distributed detection scheme. The average false alarm rates over the lifetimes of the target nodes, as illustrated in Figures 5.18 and 5.20, depict a degrading performance, with decrementing residual lifetimes of target nodes. The average false positive rate for \( N=2048 \) is 28.5\%, and the average false negative rate is 56.9\%, as compared to corresponding values of 0.4\% and 4\%, for the distributed detection scheme. This is because of the inability of the SOM-based scheme to update patterns at run-time to reflect the changing subpattern values, depicting declining energy content of the target nodes.

In Table 5.6, we compare the energy decay rates of the distributed detection scheme against corresponding values, associated with energy decay in the detector nodes of the SOM-based scheme.

<table>
<thead>
<tr>
<th>Node Deployment Density (( N ))</th>
<th>Distributed detection scheme</th>
<th>SOM-based scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GN Nodes</td>
<td>mGN Nodes</td>
</tr>
<tr>
<td>128</td>
<td>56.6</td>
<td>36</td>
</tr>
<tr>
<td>256</td>
<td>40.8</td>
<td>44.6</td>
</tr>
<tr>
<td>512</td>
<td>27.9</td>
<td>53</td>
</tr>
<tr>
<td>1024</td>
<td>20.2</td>
<td>96.1</td>
</tr>
<tr>
<td>2048</td>
<td>14.7</td>
<td>114.8</td>
</tr>
</tbody>
</table>

Table 5.6: Energy Decay Rate (\( \mu J/sec \)) comparison between the distributed detection scheme and SOM-based detection scheme for \( \alpha=0.95 \) and \( T_I=500 \).
The presence of mGN nodes in the distributed detection scheme significantly reduces the overhead associated with communication over longer distances, by individual detector nodes, towards the base station. In particular, networks with lower node deployment densities are at more advantage, as the overall distances to be traversed by the communication messages is reduced significantly. For instance, $N=128$ yields an energy consumption rate of $56.6\mu J/sec$ for a GN node, and $36\mu J/sec$ for an mGN node, whereas for a detector node of the SOM-based approach, the energy consumption rate is nearly 10 times that of the mGN node. For large values of $N$ (say $N=2048$), the higher number of GN nodes that need to communicate with each mGN node lead to higher energy utilisation rates for the mGN nodes. Nodes operating as both GN as well as mGN nodes at any given instance will have higher energy decay rates, as compared to the detector nodes of the SOM-based approach. The total number of mGN nodes operational in the network is much lesser than the number of GN nodes. Therefore, the total cost of energy consumption of the distributed detection scheme is comparatively lower than the SOM-based detection scheme, for all values of $N$.

5.5 Conclusions

In this chapter, we studied the performance of the distributed attack detection scheme proposed in Chapter 4, for variations in several algorithmic and network-level parameters, namely:

- Network traffic intensities (adversarial nodes).
- Node deployment densities.
The experimental results acquired for analysis and comparisons, were quantified in terms of the following metrics:

- Attack detection rates.
- False positive rates.
- False negative rates.
- Node energy decay rates.

As part of the scheme evaluation, we analysed the attack detection rates for variations in the total number of detector nodes in the network, as well as the network sizes and node deployment densities. Subsequently, we performed simulation experiments to study the false positive and false negative rates of the scheme. The attack detection rates show a significant increase with increasing numbers of attack detector nodes in the network. However, for smaller node deployment densities, the detection rates do not exceed 75%, even when very few packets (both attack and normal) penetrate the network. The reason for this degraded performance is the unavailability of detector nodes in several regions of the network, thus leading to the reconstruction of incomplete patterns of observed network traffic. Therefore, attack decision making is not completely accurate in such scenarios. All networks show a significant improvement in performance when fewer packets were to be analysed i.e. low TI values. The Communication phase of the attack detection scheme, when in progress, results in negligence of attack packets entering the network, at time of convergence. Therefore, lower detection rates are observable for higher traffic intensities. We infer from the study on pattern update rates that the need for having accurate pattern value updates within the detector
nodes is essential to achieve reasonable attack detection rates (Table 5.2). The performance of the scheme is severely degraded in the absence of a pattern update process in the detector nodes.

The false alarm rates of the detection scheme were compared for various node deployment densities and traffic intensities. The inability of the detector nodes to reconstruct entire patterns of traffic observations in low node density networks, led to higher false alarm rates. Higher accuracies in pattern reconstruction yielded fewer false alarm rates for larger values of $N$. We can therefore conjecture that to increase the accuracies in attack detection, higher node deployment densities are essential.

The attack detection scheme consists of both the detector (GN) nodes, as well as a subset of these nodes, operating as mGN nodes. The energy decay rate analysis shows that the rate of decline of energy content is significantly higher in the mGN nodes, as compared to the GN nodes. The added tasks of receiving attack decision packets, and further analysis, and forwarding of messages to the base station, affects the energy decay rates of the mGN nodes. Therefore, it is essential to have fewer mGN nodes operational in the network, as part of the detection process. The proposed $m$Select algorithm in Section 4.4, helps achieve this optimality for defining the total number of mGN nodes for the detection scheme.

SOM-based neural networks have been extensively used for detecting anomalous network traffic in wired high-performance networks. However, the resource-constrained nature of wireless sensor networks, accompanied with the need for regular updates of pattern values based on energy content decay, imply that such a mechanism will have low effectiveness in these networks. In the second phase of this chapter, we benchmarked the simulation results
for the mGN scheme with corresponding results obtained using a centralised SOM-based detection technique. The comparison yielded significant performance improvements of the distributed detection scheme, over the SOM-based approach. The lack of a pattern update mechanism in place for the latter led to reduced detection rates, and increased false alarm rates, with reducing target node lifetimes.

Following is an enlistment of our inferences from the simulation analysis performed:

- Increasing number of operational detector (GN) nodes in the network lead to improved attack detection rates for a given attack traffic inflow rate.

- Higher node deployment densities are required to achieve higher accuracies in pattern reconstruction, and in effect to achieve higher detection rates.

- Increasing attack intensities lead to lower detection rates for fewer number of detector nodes, whereas for networks with larger numbers of detector nodes, the increasing traffic intensities have little or no impact on the detection rates.

- The rate of update of attack threshold patterns, if not at par with corresponding energy decay rates of the target nodes, reduces the accuracy of the detection rate significantly.

- False alarm rates in the network are inversely proportional to the density of node deployment of detector nodes in the network.
• The inability of the Self Organising Map-based technique to update pattern values at network runtime, degrades its detection rate, and increases the false alarm rates significantly.
Chapter 6

Compromise-Tolerant Attack Detection Scheme

The class of malicious nodes launching a distributed denial of service attack in a wireless sensor network falls into three types, namely, injected nodes, laptop-class nodes, and compromised nodes (Chapter 3). We proposed an attack detection scheme in Chapter 4 to detect distributed flooding attacks launched by a set of malicious nodes injected into the network by the adversary-class. The scheme utilises a single GN array encompassing the entire network, to participate in the attack detection process, during each time epoch. Attack detector nodes if compromised by the adversary-class, will lead to reduced accuracy in attack detection. This is because the detector nodes constituting the GN array are required to converge by communicating with each other i.e. collaborate on a regular basis. The loss of even a small number of detector nodes of the array will lead to constitution of incomplete traffic observation patterns, in effect significantly reducing the detection rate of the scheme. In this chapter, we propose a cluster-based approach as a node
compromise-tolerant mechanism, for detection of distributed denial of service attacks in the presence of compromised sensor nodes. The cluster overlay can be imposed on any underlying network topology, for purposes of attack detection.

The aim of the scheme is to be able to operate and detect attack traffic flow in the network, even when a set of legitimate nodes in the network are compromised by the adversary-class. The set of compromised nodes may also include the attack detector nodes, responsible for acting as part of the attack detection GN array, as elaborated in Chapter 4. In this chapter we signify the need for a failure-tolerant approach for detection of distributed denial of service attack patterns. We formulate an equation to tradeoff the accuracy in the detection rate, with the cluster size, which in turn, affects the utilisation of sensor energy resources. Further, we perform a simulation analysis to test the effectiveness of our approach for attack detection in compromised node scenarios, and its superiority to the distributed attack detection scheme of Chapter 4, in terms of detection rates. Although the scheme proposed in this chapter will yield high attack detection rates for node compromise-scenarios, the comparatively higher utilisation of energy for the scheme implies that it be used only if the likelihood of having a node-compromise scenario is high. For all other scenarios, the distributed detection mechanism of Chapter 4 will suffice.

6.1 Introduction

We propose a failure/node compromise-tolerant attack pattern recognition scheme for distributed denial of service attacks in wireless sensor networks.
Cluster-based sensor networks, as defined in Chapter 3, are networks, wherein the entire network is constituted of a set of clusters, of equal distribution of nodes within each cluster. All clusters in the network have a centralised cluster head present in them, responsible for the management, data aggregation and administration of its cluster of operation. In a cluster-based topology, the cluster head communicates with the base station through a well-defined multi-hop path of intermediary nodes called data aggregation (DA) nodes. The advantage of having clusters in the network is the reduced dependence of each sensor node on a centralised base station for all their operations. The energy consumptions associated with the frequent communication of data by individual sensor nodes over longer distances in a non-cluster topology are thus overcome in such networks. The cluster heads of the network frequently monitor the status of nodes within their respective clusters, and observe for failures, incorrect readings, or low energy contents. Subsequently, the base station is informed on the status of the periodic findings of each of the cluster heads. In the event of node failure, necessary action is taken to update the cluster statistics, and if possible, arrange for replacement of the failed nodes.

Cluster-based networks also bear the advantage of failure tolerance, wherein regions of the network overlapping in the spatial distribution between two or more clusters, ensure that node losses in one cluster do not affect the flow of sensory readings from the particular region of the network, owing to the presence of other nodes in the overlapping region.

In our scheme, we propose a cluster overlay on a sensor network, with each cluster consisting of a set of detector nodes in addition to a centralised master GN (mGN) node, responsible for localised attack decision making. The purpose of having clusters of attack detector nodes for attack pattern
recognition is to facilitate tolerance to failure of detector (GN) nodes due to compromise or energy loss. The optimised cluster sizes, along with the cluster-head selection, as part of the detection scheme can be used by itself in the case where the intended underlying data delivery model of the resulting network is a cluster-based topology. On the other hand, the cluster formation can also act as an overlay on an existing network topology (flat or data aggregation), and facilitate attack detection, where otherwise all network communication associated with routine sensory operations will take place based on the underlying network routing topology.

The distributed detection scheme of Chapter 4 requires the participation of all mGN nodes in the attack detection process, at the end of every epoch of time. The loss of a single node due to compromise by the adversary-class, assuming the compromised nodes do not participate in the detection process, will lead to incomplete pattern reconstruction.

The cluster-based detection scheme requires the bifurcations of complete patterns depicting thresholds of attack patterns, defined in Chapters 3 and 4. The individual detector nodes in a cluster are responsible for collaborating with peer detector nodes within their clusters alone, for pattern reconstruction purposes. The goal of our scheme is to reduce the effect of a distributed denial of service attack, by means of timely and accurate detection. As a result, the target nodes in the network can operate for longer periods of time, implying that the frequent replacement of dysfunctional sensor nodes deployed in inaccessible environments can be avoided.

Clusters of detector nodes are formed at network initialisation time to operate independently, without the need to collaborate with detector nodes belonging to other clusters. The proposed scheme tolerates failures of detector
nodes due to energy loss or compromise, by confining the extent of damage associated with lost detector nodes, to the respective clusters of operation alone. Nodes belonging to other clusters continue uninterrupted with their task of detecting distributed attack patterns.

In Table 6.1, we define the notations for the compromise-tolerant attack detection scheme.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Number of nodes in the network</td>
</tr>
<tr>
<td>$c$</td>
<td>Number of clusters in the network</td>
</tr>
<tr>
<td>$c_{opt}$</td>
<td>Optimal number of clusters</td>
</tr>
<tr>
<td>$q$</td>
<td>Compromised node ratio</td>
</tr>
<tr>
<td>$\text{sig}_r^n(\Delta_i)$</td>
<td>Attack detection signal for a target node $r$ generated by detector node $n$ at the end of time epoch $\Delta_i$</td>
</tr>
<tr>
<td>$M$</td>
<td>Set of mGN nodes = ${mGN_1, mGN_2, ..., mGN_c}$</td>
</tr>
</tbody>
</table>

Table 6.1: Notations for the Compromise-Tolerant Attack Detection.

The contributions of this chapter are as follows:

- A cluster-based node compromise-tolerant, distributed, pattern recognition scheme for distributed denial of service attacks is proposed.

- A tradeoff equation is defined for computing the optimal cluster sizes to reduce the effects of large sized clusters on the node loss tolerance factor, and at the same time optimise on the numbers of clusters in the network, for achieving energy usage efficiency.

- A detailed simulation analysis has been done to test the effectiveness and performance of our scheme.

The chapter is organized as follows: The attack pattern model is given in Section 6.2. In Section 6.3 we present the tradeoff formulation for the optimal cluster sizes. We present the attack detection scheme in Section 6.4.
The performance evaluation of our scheme is given in Section 6.5. Finally, we enlist the concluding remarks in Section 6.6.

6.2 Attack Pattern Model

The threshold patterns for the detection scheme are defined and generated depending on the topology of the sensor network, using Equations 3.5, 3.6 or 3.7, respectively. The individual sub-patterns of a complete attack threshold pattern depict the maximum number of packets that may are receivable by the target node(s) during a given epoch of time of length $\Delta_{opt}$, from a given region of the network. The optimal time epoch length is computed using Equation 4.8, for a given value of $\alpha$ and $TI_e$.

The traffic flow from the sensor nodes of a cluster through the cluster head, and the intermediary aggregation nodes can be expressed as $f = \{f_1, f_2, ... f_{L(f)}\}$, where $L(f)$ is the length of path from node $f$ to the base station.

In Figure 6.1, we illustrate the model of a network with a set of compromised nodes, launching a distributed denial of service attack against legitimate target nodes of the network. As can be seen from the figure, the detector nodes are constituted into multiple clusters to operate in tandem to perform the attack detection process. Each cluster forms a separate GN array for localised attack detection purposes. For the example given in Figure 6.1, there are two operational GN arrays corresponding to two clusters of operation. The detector nodes belonging to a given GN array are initialised to store subpattern values for each of the $r$ target nodes in the network. These subpatterns are composed of a $(value, position)$ pair in the GN array, where...
the value field of a GN (detector) node $i$ for a target node $r$ is an integer defining the threshold $th^r_i$ of the maximum number of requests that may be accepted by node $r$ within a given time period from the region of operation of node $i$. The position fields of the GN pairs (subpatterns) identify the location of the GN node in the GN array, and is extended to incorporate cluster identification tag as well. The values for the parameters in these fields are initialised to define the structure of the GN array, i.e. successor and predecessor nodes for each GN node within each cluster $c$, at network initialization time.

Figure 6.1: A cluster-based network with a set of malicious (Compromised) sensor nodes participating in the attack.

The subpattern values generated for storage and subsequent comparison by the attack detector nodes will vary for the three network topologies. For instance, in a data aggregation-topology, the aggregation nodes in the network are responsible for accumulating the messages received from cluster heads lower in the hierarchy, and delivery to the base station. In this network, the total number of messages i.e. packets that need to be received and transmitted by a data aggregation node are much more than the number of packets that
need to be transmitted by a cluster head. Therefore, it may be seen that higher values of $d_{GN_j}(i)$ make the observed values of traffic flow towards node $j$ from GN node $i$ more significant to the detection process. Fig. 6.1 depicts a scenario, wherein the network is operating with two clusters. Traffic flow towards two example target nodes, $R_1$ and $R_2$, needs to be monitored by the detector nodes. In Table 6.2, we generate the threshold values stored by the GN nodes, for a cluster-based underlying network topology, using Equation 3.6, for a sensor network with 100 nodes, with a taxonomy, wherein the nodes are required to generate sensory readings once per second. Each row corresponds to a single threshold pattern, for a target node. The GN node ID is a two-tuple, given by $<\text{Cluster Number}, \text{GN Node Number}>$.

It can be seen from Table 6.2 that for target $R_1$, GN nodes $(0, x)$ store relatively low subpattern values, where $x \in \{0, 1, 2, 3, 4\}$. This is because fewer packets are expected from their regions towards a target node in another cluster. However, nodes belonging to the cluster 1 have higher threshold values for the same target, indicating the higher traffic intensities expected as part of the flow towards $R_1$. Similarly, GN nodes $(1, x)$ depict low threshold values for distant target node $R_2$, whereas GN nodes $(0, x)$ being in the vicinity of the target node, have larger threshold values.

<table>
<thead>
<tr>
<th>Detector ID</th>
<th>$(0, 0)$</th>
<th>$(0, 1)$</th>
<th>$(0, 2)$</th>
<th>$(0, 3)$</th>
<th>$(1, 0)$</th>
<th>$(1, 1)$</th>
<th>$(1, 2)$</th>
<th>$(1, 3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$th^{R_1}$</td>
<td>44</td>
<td>41</td>
<td>49</td>
<td>54</td>
<td>94</td>
<td>96</td>
<td>92</td>
<td>90</td>
</tr>
<tr>
<td>$th^{R_2}$</td>
<td>88</td>
<td>85</td>
<td>87</td>
<td>89</td>
<td>35</td>
<td>41</td>
<td>46</td>
<td>47</td>
</tr>
</tbody>
</table>

Table 6.2: Threshold (sub-pattern) values for target nodes $R_1$ and $R_2$. 

\[202\]
6.3 Optimal Cluster Size

Large numbers of clusters in the network will tolerate node compromise to a larger extent. Similarly, smaller numbers of clusters will lead to improved resource usage, at the cost of reduced tolerance to node compromise or failure, and reduced accuracy in attack detection. For a network with a single cluster (Chapter 4), even the loss of a few detector nodes will reduce the accuracy of attack detection. On the contrary, having a very large number of clusters will lead to more numbers of cluster heads participating in the detection process, thus increasing the energy resource usage.

For a network of size $N$, with $c$ clusters, the total number of nodes excluding the cluster head in a cluster are given by: $n = \frac{N}{c} - 1$, which also defines the number of GN nodes operational in the network. If the value of $c$ is large, the overall energy consumption rate of the scheme will be very high, due to the added cost of cluster head operations. Similarly, having a small number of clusters will reduce the success rate of the detection scheme. Therefore, an optimal value of $c$ must be found to achieve the goal of reducing the energy consumption of the network for a reasonable degree of accuracy in attack detection.

We define an analytical model to determine the optimal number of clusters, $c_{opt}$, for a given network with given dimensions and node deployment densities. The network area is assumed to be a square grid with side $a$. The total number of grids in the network is equal to the number of clusters, $c$. For a network with $N$ nodes, where $N =$ total nodes in network, the average number of nodes per cluster is $\frac{N}{c} - 1$, and the total number of cluster heads is $c$. It is also assumed that each cluster head is located approximately in
the center of its grid. A typical cluster-based sensor network in the form of a square grid of side \( a \), is illustrated in Figure 6.2.

![Figure 6.2: Square grid network with side = \( a \) and number of clusters = \( c_{\text{opt}} \).](image)

For a network with \( N \) nodes, and side of grid = \( a \), the expected distance from the cluster head to individual nodes in the cluster is given by:

\[
E[d_{CH}] = 0.707a \sqrt{\frac{c}{c}}
\]  

(6.1)

In a scenario without node loss due to compromise and/or failure, the total energy consumed by the network is given by:

\[
E_{\text{total}} = c(E_{CH} + (\frac{N}{c} - 1)E_{\text{nodes}})
\]  

(6.2)

where, \( E_{CH} \) is the energy consumed by a cluster-head, and \( E_{\text{nodes}} \) is the energy consumed by the non-cluster head nodes in a cluster. The cluster heads aggregate data and follow a multi-hop route for data delivery to the base station. For shorter communication distances, the Friss free-space model is used to model the power loss (power loss is the square of the inter-node distance \( d^2 \)), whereas for longer distances (between the cluster head and the Base Station), the multi-path fading model is used, where the power loss is
the fourth power of the distance ($d^4$) (Kim et al., 2005). The value of $E_{CH}$ is a function of energies utilised for the following processes: receiving data from the $n$ nodes, aggregation of data from $n$ nodes ($E_{DA}$) and data transmission to the base station is given by: $E_{BS} = E_{elec} + \epsilon_{mp}(d_{BS}^4)$.

$\epsilon_{mp}$ is the energy utilised for transmission of a bit of message over a given distance (multi-path fading model). $E_{elec}$ is the electronics energy, which depends on operations such as digital coding, modulation, filtering, and spreading of the signal (Kim et al., 2005). The energy consumed by a cluster-head for an $l$-bit message is given by:

$$E_{CH} = l.[n.E_{elec} + n.E_{DA} + E_{elec} + \epsilon_{mp}(d_{BS}^4)]$$ (6.3)

Since the non-cluster head nodes are in vicinity of the cluster head, we consider the energy consumption to follow the Friis free-space model, where the power loss is a square of the distance between the nodes and the cluster head ($d^2_{CH}$). The non-cluster head nodes only transmit data to the cluster head, and therefore the energy consumption of these nodes is given by:

$$E_{nodes} = l.[E_{elec} + \epsilon_{fs}(d_{CH}^2)]$$ (6.4)

where, $\epsilon_{fs}$ is defined as the energy utilised per bit of message transfer using the free-space model. We define $q$ as the ratio of nodes lost due to compromise by an adversary. We also define $\gamma = 1 - q$.

For a node-compromise scenario, assuming the abstention of the compromised sensor nodes belonging to a cluster, from participating in the pattern reconstruction process, the total energy utilisation is given as follows:
\[E_{total} = c \cdot [E_{CH} + \gamma \cdot n \cdot E_{nodes}]
\]

\[= l \cdot [(N - c)E_{elec} + NE_{DA} + cE_{elec} + c\epsilon_{mp}(d_{BS}^4) + \gamma NE_{elec}]
+ \frac{\gamma N\epsilon_{fs}(0.707a)^2}{c} - \gamma cE_{elec} - \gamma \epsilon_{fs}(0.707a)^2\]  \hspace{1cm} (6.5)

The total energy utilisation is minimised by equating the first derivative of Equation 6.5 to zero, and deriving the value of \(c_{opt}\). The optimal number of clusters in the presence of compromised nodes is thus given by:

\[c_{opt} = \sqrt{\frac{\gamma N\epsilon_{fs}0.499a^2}{\epsilon_{mp}(d_{BS}^4) - \gamma E_{elec}}}\]  \hspace{1cm} (6.6)

In Fig. 6.3, we have plotted the values of \(c_{opt}\) for varying numbers of nodes in the network, and varying node compromise ratios. The values of the constants were taken as: \(l=1\) byte, \(\epsilon_{fs}=10\) pJ/bit/m², \(\epsilon_{mp}=0.0013\) pJ/bit/m², \(E_{elec}=50\) nJ/bit, \(E_{DA}=5\) nJ/bit/signal, \(a=100\) and \(d_{BS}=\) variable (average distance from a cluster-head to a base station for a 100 x 100 network with number of nodes = \(N\)).

Figure 6.3: Optimal Number of Clusters vs. Number of Nodes for varying values of \(\gamma\).
The total number of clusters in the network do not increase linearly with increasing values of $N$. For instance, $N=128$ will yield a $c_{\text{opt}}$ value of 18 for $\gamma=1\%$, whereas for $N=2048$, $c_{\text{opt}} = 84$. This non-linearity helps understand the tradeoff that is achieved from Equation 6.6. Linear increases in the number of cluster-heads, for corresponding increase in the node deployment densities will lead to more number of nodes operating as cluster-heads, and sustaining high energy usage rates. Equation 6.6 thus computes a balanced number of cluster heads in the network, under varying adversarial node presence, so as to reduce the overall energy consumption rates incurred on the sensor nodes by the detection scheme.

### 6.4 Attack Detection Scheme

The attack detection scheme proposed in this section generates and maintains clusters of detector nodes in the network. Each cluster is autonomous in nature, and performs its attack detection process without the need to collaborate with detector nodes belonging to the other clusters. The purpose of having clusters for pattern recognition is to facilitate tolerance to failure of detector nodes due to compromise or energy loss. The cluster-based detection algorithm requires bifurcations in the complete pattern depicting an attack, based on the cluster ID. The detector nodes in a cluster are thus responsible for collaborating with peer detector nodes within their clusters alone, for pattern reconstruction purposes. In the event of node compromise or failure, the extent of damage is confined only to the particular cluster in question, with the remaining clusters operating unfazed.
The detector nodes detect patterns, and have a nominated master GN (mGN) node within each cluster, responsible for localized decision making within the cluster. The mGN node can be any node in the cluster, and is selected randomly by the base station at network initialisation time.

The attack detection scheme as illustrated in Algorithm 6.1, consists of the following five phases of operation:

Cluster Formation

The base station generates \( c_{opt} \) number of clusters in the network based on Equation 6.6. Each node, based on its location coordinates, is assigned to a particular cluster, and the corresponding cluster-head is informed about the list of all nodes belonging to its cluster. The cluster-heads are selected by the base station based on a uniform probability distribution. The base station also selects one random detector node in every cluster to operate as a master GN (mGN) node. In addition, each node in the network is assigned a node ID based on its location and cluster of operation. This ID, along with the total number of nodes within each cluster, are required details for each GN node to locate its peer GN nodes, for pattern reconstruction purposes. All nodes within each cluster of operation participate in the attack detection process, and belong to the GN array.

As detailed out in Algorithm 6.1, each detector node is initialised with two tables in its local memory, namely, traffic flow observation table and threshold table. The detector nodes store the maximum threshold value, \( th_n^r \), associated with each of the \( r \) targets within their respective threshold tables. The traffic flow observation table has constantly changing values based on the observation of the neighborhood traffic flow towards the \( r \) targets. A comparison
between corresponding values for a given target \( r \) in these two tables at the end of a given time epoch \( \Delta_{opt} \), decides the output signal \( \text{sig}^n_r(\Delta_{opt}) \) to be generated by the detector node \( n \) for transmission to its designated mGN node. The addition of new targets and deletion of existing ones is done by the base station at network initialisation time. The value of \( \Delta_{opt} \) is computed using Equation 4.8, with \( m = 1 \) and \( n = \frac{N}{c} - 1 \).

**Observation**

This phase of operation is the same as the one given in Algorithm 4.1, wherein each GN node promiscuously monitors packets initiating or transiting through its neighborhood towards one of the \( r \) critical target nodes. The GN nodes update their traffic flow table accordingly. Each GN node will store \( r \) threshold subpattern values, one for every target node of the network. The fixed length of each epoch of time facilitate the synchronisation of the messages that are exchanged within each cluster of the network.

**Communication**

In this phase, within each cluster \( c \), all detector nodes communicate with exactly two other adjacent nodes, namely, the successor (\( nsucc \)), and the predecessor (\( npred \)) nodes to facilitate reconstruction of complete traffic patterns from individually observed subpatterns of traffic flow. Consequently, a complete traffic flow pattern for each of the given target nodes \( r \) for a given time frame is generated at each mGN node.

Each detector node shares a pairwise distributed key with three other nodes, namely, successor, predecessor and the masterGN (mGN) node, and therefore, will store three pairwise keys in its memory. The mGN nodes store
1.a Cluster Formation
for \( i = 1 \) to \( c \) do
| pick random decision-making GN node \((mGN_c)\)
end

1.b Pattern Generation & Learning
for \( i = 1 \) to \( n \) do
| Generate pattern: \( p_n = \{p^1_n, p^2_n, ..., p^r_n\} \)
end

2. Observation
for \( i = 1 \) to \( n \) do
  for \( j = 1 \) to \( r \) do
    Monitor traffic flow towards \( r \) targets and Update traffic flow
    observation table locally
  end
end

3. Communication
for \( i = 1 \) to \( n \) do
  for \( j = 1 \) to \( r \) do
    if traffic flow table entry for \( r > th^c_r \) then
      Communicate with neighboring nodes \( n_{succ} \land n_{pred} \) to
      reconstruct subpattern \( \{p^r_{n_{succ}}, p^r_{n_{pred}}\} \)
    end
  end
  for \( i = 1 \) to \( c \) do
    \( mGN_i \) receives \( r \) observations from \( \frac{N}{c} \)-1 GN nodes, during \( \Delta_i \)
    for \( i = 1 \) to \( r \) do
      Generate decision signal: \( attack^c_r \) or \( normalcy^c_r \)
    end
  end
end

4. Verdict
for \( i = 1 \) to \( c \) do
  \( \forall r \), if \( attack^c_r = 1 \), Transmit \( attack^c_r \) to base station
end

5. Pattern Update
for \( i = 1 \) to \( n \) do
  Update \( th^c_n \)
end

Algorithm 6.1: Cluster-based Distributed Denial of Service Attack Detection Scheme.
all keys that they share with the detector nodes in their cluster, as well as a
key that they share with the base station. All messages exchanged between
GN/mGN nodes/base station, will have the following format:

\[ A \rightarrow B : m, MAC(K_A^B, m, ctr(\Delta_i)) \]

where,

\[ A = GN/mGN \text{ Source Node} \]
\[ B = < \text{nsucc}(A) \lor \text{npred}(A) \lor q_A \lor \text{base station}> \]
\[ ctr(\Delta_i) = \text{counter value as a function of the current time epoch } \Delta_i \]

At each detector node, if the number of incoming requests for a particular
target \( r \) during the current time epoch exceed the stored threshold \( th_r \) value in
the pattern table, and its successor and predecessor nodes have also detected
similar anomalies given by their respective sub-patterns, \( p^r_{nsucc} \) and \( p^r_{npred} \), the
detector node \( n \) will generate an attack \( r \) signal for the current time epoch.
On the contrary, a normalcy \( r \) signal generated by the GN nodes implies
incomplete or no-match between the observed traffic pattern and the stored
pattern of anomalous behavior for traffic destined for node \( r \).

All detector node communication takes place in parallel, and therefore, the
large numbers of detector nodes do not significantly affect the overall com-
munication delay of the scheme. After comparison with the adjacent detector
nodes, the outcome of a pattern recognition process from each individual
detector node is communicated to its designated mGN node.
Verdict

At the end of the current time epoch \( \Delta_i \), exactly half of the detector nodes within each cluster \( c \) communicate with the designated mGN node, to convey their respective observations. Neighbouring GN nodes alternate in consecutive time epochs, in communicating with the mGN node of a cluster, so as to reduce the duplication of messages, and in effect, reduce the energy utilisation rates associated with the attack detection process. If the number of GN attack signals within a given cluster = \( \frac{N_c - 1}{2} \), for any or all of the specified targets arriving at the \( mGN_c \) for cluster \( c \), the traffic flow is classified as an attack, i.e. \( \forall r \), if \( \prod_{i=1}^{\frac{N_c - 1}{2}} \text{attack}^{c}_{r}(i) = 1 \Rightarrow \text{an attack against } r \text{ is declared to be in progress.} \)

Pattern Update

The pattern update rate is modeled based on Equations 5.5, 5.6 and 5.7, for each of the three network topologies respectively. Upon successful confirmation of an attack signal, the base station sends a signal to induce node \( r \) into sleep mode for a finite period of time. Subsequently the base station ensures that if alternative resources are available, they are sent a signal to designate them the task of continuing with the sensing operations from the region of operation of node \( r \). For instance, if node \( r \) belonged to a data aggregation-based topology, responsible for aggregation of received data, the base station sends a request to another active node available within the vicinity of node \( r \) instructing it to take over the data aggregation responsibilities of \( r \).
6.5 Evaluation

In this section we provide a detailed analysis of the simulations that were performed for varying network and algorithmic parameter values. The purpose of our experiments was to test the following hypothesis:

- Detection rate will be maximum for $c = c_{opt}$.
- Compromise of detector nodes will have a receding impact with decreasing cluster sizes.
- Fewer detector nodes need to operate for networks with lower deployment densities.
- Increasing node densities will lead to improved detection rates.

6.5.1 Experimental Setup

The deployment of sensor nodes follows a uniform distribution throughout the network. The deployment region is a square grid with side $a$. We assume that all nodes are equally likely candidates for loss owing to failures, battery exhaustion, or compromise. In addition to their routine sensing operations, the attack detector nodes also participate in the attack detection process. It may therefore be safely presumed that the task of selectively identifying and launching attacks against the detector nodes by an adversary is nontrivial.

Sensor nodes are assumed to have a single interface for both transmit and receive operations. We considered a standard sensor node with average energy consumptions for transmission $E_{trans} = 100 \text{ nJ}/\text{bit}$ and $E_{recv} = 50 \text{ nJ}/\text{bit}$, with the maximum radio range of each sensor node being 50 meters (Krohn et al., 2006).
6.5.2 Simulation Parameters

The following parameters were selected for the simulation setup:

- $SR$: The transmission range of a sensor node $\sim 50m$.

- $\Delta_{opt}$: Time epoch length for the number of mGN nodes, $m=\frac{N}{c_{opt}}$ and number of GN nodes, $n=N-m$.

- $TI$ (Traffic Intensity): Packets generated towards the $r$ target nodes during a given time epoch ($\Delta_{opt}$).

- $TI_e$ (Traffic Intensity): Packets generated towards the $r$ target nodes during a given time epoch ($\Delta_{opt}$), in terms of energy usage by the target nodes.

- $c_{opt}$: Optimal cluster size for a network with given dimensions.

- $q$: Percentage of compromised nodes.

6.5.3 Analysis

In order to set the thresholds in the attack patterns, the network is trained by testing it with varying values of traffic intensities (both attack and normal). In addition, the time epoch length, $\Delta_{opt}$, is generated for $\alpha = 0.95$, and the corresponding traffic intensity, $TI$, is taken as 500 packets/$\Delta_{opt}$ implying that the expected numbers of packets by each of the $r$ target sensor nodes during $\Delta_{opt}$ is 500. It may be noted here that in a real-world scenario, to increase the longevity of the sensor nodes, the value of $TI$ needs to be less. We also assume that all nodes belonging to a cluster, $n = \frac{N}{c} - 1$, participate in the attack detection process.
Detection Rate vs. Node Loss Ratio

We study the attack detection rates for variations in the value of $q$. As mentioned earlier, the distributed denial of service attack detection scheme proposed in Chapter 4 is intended for adversary-injected nodes in the network, and does not tolerate failures associated with node compromise. The scheme proposed here utilises multiple clusters operating in parallel, to detect distributed denial of service attack patterns, to ensure that node losses up to a certain extent do not significantly affect the detection success rate.

In Figure 6.4, the detection rate reaches nearly 34% for $q = 1\%$ and $c = c_{\text{opt}}$. For the same value of $c$, about 27% of the attacks are detected even when node loss ratio $q$ reaches 15%. For smaller values of $c$, 0.2$c_{\text{opt}}$ and 0.5$c_{\text{opt}}$, the detection rate varies from 28% for $q = 1\%$, to 24% for $q=15\%$. Fewer numbers of clusters (small $c$) lead to reduced tolerance to node loss, and therefore lower the overall detection rate. The detection rate reaches 0% for nearly 30% node loss for smaller values of $c$. However, for larger values of $c$, the network, being more tolerant to node failure, demands 50% node loss before reaching the zero success rate point.

In Figure 6.5, the detection rate shows a significant improvement as compared to the 128 node case. With $c = c_{\text{opt}}$ and $q=1\%$, the detection rate is nearly 80%, and reduces to 61% for $q=15\%$. The improved success rate here is due to the higher densities of node deployment, which ensures that more number of nodes participate in the detection process, with added tolerance to node failure.

Similarly, Figure 6.6 shows further improvements in the detection rates for increasing numbers of clusters. For $c = c_{\text{opt}}$ and $q = 1\%$, the detection rate is
Figure 6.4: Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=128$). A peak detection rate of 34% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=50\%$, the detection rate becomes negligible for all cluster sizes.

Figure 6.5: Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=256$). A peak detection rate of 80% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=50\%$, the detection rate becomes negligible for all cluster sizes. The detection rates for $c=c_{opt}$ and $c=0.5c_{opt}$ are comparable.
nearly 85%, and for \( q = 15\% \), is around the 68%. The attack detection process is thus reasonably accurate even when 15% of the nodes in the network are lost.

![Figure 6.6: Detection Rate vs. Node Compromise Ratio (q) for Varying Cluster Size (N=512). A peak detection rate of 85% is observable for \( q=1\% \), \( c=c_{opt} \). For less than \( q=67\% \), the detection rate becomes negligible for all cluster sizes.](image)

For networks with higher node deployment densities, Figures 6.7 and 6.8, the detection rate is nearly 94% when all \( N \) nodes are operational, and \( c = c_{opt} \), and is nearly 72% when \( q \) is 15%. It may be noted here that even when \( c = 0.5c_{opt} \), the detection rate is very high. This implies that for larger \( N \), after a certain number of clusters are operational in the network, the role of increasing values of \( c \) in improving the attack detection rate is diminished. In such scenarios, lower \( c \) values can achieve high detection rates, with the advantage of lower scheme convergence delays, and at the cost of slightly higher energy decay rates.

From the analysis of the experiments, it can be concluded that the optimal value of \( c \) given by \( c_{opt} \) yields high success in attack detection for smaller values
Figure 6.7: Detection Rate vs. Node Compromise Ratio (q) for Varying Cluster Size (N=1024). A peak detection rate of 94% is observable for q=1%, $c=c_{opt}$. For less than q=70%, the detection rate becomes negligible for all cluster sizes.

The attack detection process relies on parallel inter-node communication for collaboration and pattern reconstruction purposes (see Algorithm 4.1). The increasing densities of node deployment in the network will increase the total numbers of nodes in each cluster of the network, and thus aids in significantly increasing the tolerance of the detection scheme to failure. As a result, a markable increase can be observed in the success rate of the attack detection process. However, this improved success rate is subdued by the increasing value of q. As can be seen from Figure 6.9, increasing values of N lead to improved success rates in attack detection. For $c = c_{opt}$ and $q = 10\%$, the detection rate is nearly 81%, when the network has 2048 nodes, and is nearly 31% when the network has 128 nodes. For $c=0.1c_{opt}$, the detection rate increases from nearly 12% for $N=128$, to 60% for $N=2048$. Thus the
Figure 6.8: Detection Rate vs. Node Compromise Ratio ($q$) for Varying Cluster Size ($N=2048$). A peak detection rate of 97% is observable for $q=1\%$, $c=c_{opt}$. For less than $q=70\%$, the detection rate becomes negligible for all cluster sizes.

The attack detection rate shows a steady improvement for increases in the total numbers of clusters in the network.

Figure 6.9: Detection Rate vs. $N$ for $q=10\%$. A peak value of 80% is observable for $N=2048$.

**Observation 6.1:** Larger $N$ will increase tolerance to node loss more than smaller $N$, and therefore higher node deployment densities improve the overall effectiveness of the attack detection scheme.
False Alarm Rates

In Figure 6.10, we illustrate the false positive rate of the attack detection scheme for variations in the node compromise ratio \(q\), and \(c=c_{opt}\), \(TI=500\).

![False Alarm Rates](https://via.placeholder.com/150)

Figure 6.10: False Positive Rate vs. Node Deployment Density \(N\) for varying Node Compromise Ratio \(q\). A peak value of 32% is observable for \(q=70\%\) and \(N=128\).

The false positive rates are lower for high density networks for smaller values of \(q\), due to the accuracy in the pattern reconstruction process, achieved after the observations from a higher number of nodes are used for generating a verdict signal by the mGN nodes. In less dense networks, the overall detection rate is reduced due to the smaller sized observation patterns that are generated from fewer numbers of nodes. In turn, the likelihood of having incomplete patterns reconstructed by the detector nodes is higher. However, with increasing numbers of compromised nodes in the network i.e. higher values of \(q\), the false positive rates reach their maximum values, and no significant variation is observable in these values, regardless of the node deployment densities. This is because the loss of even a few number of nodes in the network will lead to reconstruction of incomplete patterns. Therefore, the
impact of a high degree of node compromise nullifies the advantage of higher
detection rates, associated with more deployed detector nodes.

In Figure 6.11, we illustrate the false negative rate of the scheme, for varying $q$. The false negative rate depends on the property of the detection scheme, which demands regular convergence of the scheme at the end of each time epoch. All attack packets penetrating the network at time of scheme convergence remain unnoticed by the detector nodes, and are tagged as false negatives. As seen from the figure, the false negative rate is higher for networks with low node deployment densities. This is because of the absence of detector nodes in certain regions of the network, increasing the likelihood of having higher numbers of unobserved attack packets.

![Figure 6.11: False Negative Rate vs. Node Deployment Density $N$ for varying Node Compromise Ratio ($q$). A peak value of 68% is observable for $q=70\%$ and $N=128$.](image)

For $q=1\%$, the false negative rates are higher, as compared to corresponding values for a non-cluster approach (Figure 5.13). This is because, the overall delay in convergence of a larger number of GN arrays, associated with more number of clusters, in such networks, increases the likelihood of attack packets penetrating the network, unnoticed.

221
Both the false positive and the false negative rates are higher for larger networks, with low node deployment densities. These rates reflect on the need for having a complete coverage of an entire network, so as to facilitate the reconstruction of accurate patterns of observed network traffic flow, by the GN and the mGN nodes. The presence of large numbers of detector nodes in such networks, helps achieve higher accuracies in pattern reconstruction, in turn reducing the false alarms in the network. However, with increasing numbers of nodes compromised in the network, the false alarm rates taper for all node deployment densities, thus nullifying the effect of attack detection.

**Energy Decay Rate**

In Table 6.3, we illustrate the overall energy consumption rates of the cluster-heads of the network per unit of time. As expected, the optimal cluster number \(c_{opt}\) shows the lowest energy consumption rates for all values of \(N\). Increasing node deployment densities lead to increase in the total energy consumption rates. However, it may be observed from the table that for \(N=2048\) and \(c = c_{opt}\), the energy consumption rate is not a 10x multiple of the \(N=256\) case. This is because for smaller \(N\), larger distances need to be traversed by the messages exchanged between the nodes, and therefore longer delays. In addition, the total number of messages exchanged in the \(N=256\) scenario will be much less than when \(N=2048\). Therefore, the energy loss associated with increasing values of \(N\) is compensated with corresponding reductions in the inter-node distances for increasing \(N\).

From the simulation results it may be deduced that optimal values of \(c\) may not be the most appropriate setting in certain network conditions. For smaller values of \(N\), if only half the number of cluster-heads are operational,
the number of nodes affected with the additional cluster-head operations is halved, but at the cost of degraded attack detection rates. For instance, the energy consumption rates for $N=128$ for $c=0.5c_{\text{opt}}$ is equal to 52µJ/sec, which is very close to 43µJ/sec for the $c=c_{\text{opt}}$ case. In mission-critical environments, wherein the accuracy in attack detection is more significant, higher energy utilisations in more numbers of nodes can be tolerated, and therefore $c = c_{\text{opt}}$ is the most appropriate cluster size in such scenarios.

**Effectiveness of Cluster-based Detection**

In this subsection we compare the performance of the distributed attack detection scheme of Chapter 4, with the cluster-based approach defined in this chapter. The purpose of the comparison is to strengthen the argument that having multiple clusters operational in the network, is essential in achieving high detection rates, when sensor nodes are vulnerable to node compromise attacks. As illustrated in Figure 6.12, the highest detection rate achieved when a single cluster is operational is 45%, for $N=2048$ and $q=0.5\%$, whereas for lower values of $N$, the detection rate is even less. The graph tapers with increasing node compromise ratio, with the detection rate reaching 0% when $q=70\%$. The detection scheme does not tolerate the presence of compromised

<table>
<thead>
<tr>
<th>$N$</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05$c_{\text{opt}}$</td>
<td>509</td>
<td>511</td>
<td>1025</td>
<td>1370</td>
<td>1653</td>
</tr>
<tr>
<td>0.1$c_{\text{opt}}$</td>
<td>255</td>
<td>342</td>
<td>415</td>
<td>692</td>
<td>841</td>
</tr>
<tr>
<td>0.2$c_{\text{opt}}$</td>
<td>104</td>
<td>151</td>
<td>215</td>
<td>332</td>
<td>463</td>
</tr>
<tr>
<td>0.5$c_{\text{opt}}$</td>
<td>52</td>
<td>76</td>
<td>110</td>
<td>165</td>
<td>230</td>
</tr>
<tr>
<td>$c_{\text{opt}}$</td>
<td>43</td>
<td>63</td>
<td>92</td>
<td>134</td>
<td>196</td>
</tr>
</tbody>
</table>

Table 6.3: Energy Utilisation Rates for Cluster-Heads(µJ/sec)
nodes in the single operational GN array. Therefore, very low detection rates are observable.

![Distributed Detection without Clusters](chart)

Figure 6.12: Attack Detection Rate vs. Node Compromise Ratio \((q)\) for varying Node Deployment Densities \((N)\) and \(c=1\). The peak detection rate is 45\% for \(N=2048\) and \(q=1\%\). The detection rate reaches zero for all \(N\), when 70\% of the nodes are compromised.

In Figure 6.13, we illustrate corresponding detection rates for the cluster-based detection scheme. As can be observed, the individuality of patterns belonging to each GN array i.e. cluster, help tolerate faults and/or node compromises, to a greater extent than the single-cluster approach above. The detection rate is nearly 98\% for \(N=2048\) and \(q=1\%\), yet again emphasizing the need for having more nodes in the network, to achieve higher success in detection. The scheme performs reasonably well even when nearly 15\% of the nodes are lost due to compromise or failure, with the detection rate being 61\% for \(N=256\), as compared to 32\% for the single-cluster scenario. A similar trend can be observed for all other values of \(N\).

In Table 6.4, we compare the energy decay rates and the number of cluster-heads and mGN nodes for \(\alpha=0.95\) and \(\gamma=10\%\). As can be inferred from the table, increasing number of nodes in the network will lead to an increase in
Figure 6.13: Attack Detection Rate vs. Node Compromise Ratio ($q$) for varying Node Deployment Densities ($N$) and $c=c_{opt}$. The peak detection rate is 96.5% for $N=2048$ and $q=1\%$. The detection rate is very low for all $N$, when 70% of the nodes are compromised.

both the number of cluster-heads as well as the number of mGN nodes, of the cluster-based and the mGN-based schemes, respectively. However, the rate in increase in the value of $||M||$ is much less than that of $c_{opt}$. For instance, for $N=2048$, a total of 93 clusters are operational in the network, whereas for the same value of $N$, only 32 mGN nodes operate. This is because, the mSelect algorithm operates by ensuring that a minimal set of mGN nodes are selected to ascertain connectivity of each GN node to at least one node from the set $M$ of mGN nodes, with no significance given to any failure or compromise-tolerance aspects. On the contrary, in the cluster-based detection scheme, the tradeoff formulation for calculating the optimal number of cluster-heads in the network, for a given compromise node ratio, ascertains a reasonable accuracy in the attack detection process in the presence of compromised nodes, achieved through inter-node collaboration within individual clusters, by optimising on the energy utilisation incurred by the detection
scheme. Thus, having a large number of cluster-heads (e.g. $c=c_{opt}$) comes at the cost of more energy resource utilisation of the cluster-heads.

| N    | $c_{opt}$ | Energy Usage ($\mu$J/sec) | $||M||$ | Energy Usage ($\mu$J/sec) |
|------|-----------|--------------------------|--------|---------------------------|
| 128  | 23        | 43                       | 17     | 36                        |
| 256  | 33        | 63                       | 18     | 44.6                      |
| 512  | 46        | 92                       | 19     | 53                        |
| 1024 | 61        | 134                      | 21     | 96.1                      |
| 2048 | 93        | 196                      | 32     | 114.8                     |

Table 6.4: Comparison of Total Cluster-Heads and Total mGN Nodes and corresponding Energy Decay Rates.

The presence of compromised nodes in the network demands the need for a failure-tolerant approach towards distributed denial of service attack detection. We can therefore infer that multiple clusters of detector nodes if operational in tandem, will help achieve higher detection rates in such scenarios, at the cost of degraded energy resource utilisations, as compared to the cluster-less scheme. It is therefore recommended to use the cluster-based scenario only in environments where node compromise attacks can be anticipated.

### 6.6 Conclusions

In this chapter we defined a cluster overlay, as a node compromise/failure-tolerant mechanism, for detection of distributed denial of service attack patterns. The cluster-based attack detection scheme is designed to operate as an overlay on an underlying network topology, or as a routing topology in itself,
wherein the cluster-based approach for data delivery to the base station can be adhered to by all the sensor nodes in the network.

The scheme initialises by performing clustering of the attack detector nodes into a set of optimal clusters, defined as $c_{opt}$. The optimal cluster sizes are computed based on a tradeoff equation, formulated to optimise between the attack detection rates and the overhead associated with the scheme in terms of energy utilisation rates of the detector nodes, under variation of the number of attacker nodes ($q$). Larger values of $c_{opt}$ will reduce the pattern lengths within each cluster i.e. reduce the number of nodes collaborating within each cluster, thus leading to higher false alarm rates, and lower attack detection rates. The energy usage associated with higher numbers of clusters is lesser for larger $c_{opt}$ values. Smaller values of $c_{opt}$ will reduce the tolerance of the scheme to node compromise, and will lead to degraded attack detection rates, as is observable from Figure 6.9. We performed simulation experiments to test the effectiveness of our scheme for varying algorithmic and network-level parameters. In particular, we studied the attack detection rates, energy decay rates and false alarm rates of the scheme for variations in the node compromise ratio and $c_{opt}$ values.

We performed a comparison of a single-cluster approach (Distributed detection scheme of Chapter 4) with the cluster-based scheme, in the presence of compromised nodes. The cluster-based scheme performed significantly well in the presence of compromised nodes, thus implying that there exists a need for having multiple clusters operational in the network, to achieve higher success in attack detection, in the presence of compromised nodes. However, the cluster-based scheme displayed increasing energy utilisation rates for increasing numbers of nodes in the network. Comparatively, the distributed
detection scheme of Chapter 4 has decreasing energy utilisation rates with increasing mGN nodes in the network, without compromising the attack detection rates. The attack detection rates do not degrade significantly because of the collaborative nature of the algorithm, to operate as a single GN array, as compared to multiple individual GN arrays operating in the scheme proposed in this chapter. The tradeoff formulation for calculating the optimal number of cluster-heads in the network compromises energy conservation to ensure that a reasonable accuracy in the attack detection process is achieved through inter-node collaboration within individual clusters. To avoid unnecessary overhead associated with the operation of a large numbers of clusters in the network, it can therefore be conjectured that the proposed cluster-based approach for attack detection is used only when the threat of a node compromise attack exists. The distributed attack detection scheme of Chapter 4 would suffice for the other scenarios.
Chapter 7

Conclusions

7.1 Summary

The availability of sensor nodes is under constant threat from Distributed Denial of Service attacks. In the initial phase of this thesis, we modeled distributed denial of service attacks *aka* energy exhaustion attacks, in wireless sensor networks. The purpose of attack modeling was to ascertain that appropriate attack detection approaches are subsequently proposed for detecting such attacks in a timely and energy-efficient manner. Moreover, the detection of such attacks is the first step towards any counter-measures, including mitigation, that may be necessary for appeasing the effects of the attack upon achieving success in attack detection.

The resource-constrained nature of sensor nodes demands the presence of light-weighted, in-network, distributed, and scalable mechanisms for detection of malicious attacks, distributed denial of service, or otherwise. We elaborated on the need for novel attack detection techniques, considering that the attack detection techniques proposed for such attacks in high performance
networks, due to their resource demanding nature, are impractical for unaltered deployment on resource constrained sensor networks. The lack of a gateway, as a single point of entry into the network increases the vulnerability of such networks, and further complicates the attack detection process. We defined the attack model, to illustrate that such attacks cannot be successfully detected by a single detector node in the network. We also modeled the overhead incurred on the adversary-class, when launching such an attack, to prove that a distributed version of such an attack can prove to be more successful than a centralised one.

The adversarial nodes were classified into three categories, namely, injected nodes, laptop-class nodes and compromised nodes. The classification was done based on the capabilities of these nodes. In Section 2.2, we described various existing attack models for wireless sensor networks, and elaborated upon how these attacks can culminate into distributed denial of service attacks. In particular, we defined the probability of having any of these attacks culminate into distributed denial of service attacks, and the need for having colluding adversaries participating in the attack process, for achieving higher success.

Subsequently, we modeled the attack detection process as a pattern recognition problem, and emphasized on the need for having a distributed pattern recognition mechanism in place, to achieve success in attack detection, without incurring significant overhead on the limited energy resources of the sensor nodes. In particular, a network model was defined to classify wireless sensor networks into topologies, based on the source-sink data delivery model. Three distinct classes of wireless sensor networks were defined, and the traffic
flow, inclusive of both attack and normal traffic, was defined for each topology, separately. We also classified a set of legitimate sensor nodes as target (victim) nodes in each topology, based on the significance of the nodes to the network operations. We modeled the attack based on the expectation that an attack launched by the adversary class against these target nodes will prove to be more disruptive. We proposed a model for expected traffic flow towards the victim node set, based on several criteria, namely, the node deployment densities, proximity of the target nodes to the base station, and the proximity of the target nodes to the detector nodes. These parameters facilitate generation of a sequence of threshold subpattern values, that depict bounds on the maximum traffic flow, permissible towards a given target node, during a fixed interval of time. The defined threshold values are stored in the $d$ attack detector nodes of the network. All detector nodes also maintain a traffic observation table, defined in Section 3.5, in their local memory. This table is updated with the observed traffic flow towards the victim node set in each frame of time, $\Delta$. Subsequently, the updated values from the traffic observation table are compared with the previously generated traffic threshold subpatterns, to decide on whether the observed traffic can be labeled as anomalous in nature, or not.

A single traffic observation value will not generate a conclusive decision on an attack. However, a complete reconstruction of the traffic observation pattern, constituting of these subpattern values, will facilitate in the decision making process. We illustrated the need for having multiple sensor nodes, with added responsibilities, to detect such attacks, when launched from multiple ends of the network, by the adversarial nodes. We analysed the energy resource usage associated with the launch of these attacks, by the adversary
class, and concluded that if the attacks are launched in a distributed manner, from multiple ends of the network, they will prove to be more successful, as compared to scenarios where the attack is launched from a single front, by a single adversarial node.

In Chapter 4, we proposed the attack detection scheme consisting of five phases of operation, to be executed sequentially within each epoch of time, of length $\Delta_{opt}$. Distinct topology-based threshold patterns for each of the $r$ target nodes in the network are generated for comparison with actual traffic flow observations by the attack detector (GN) nodes. Subpatterns of threshold values depicting distributed flooding attacks against the target node set $T$ are generated based on the criteria, defined in Chapter 3, and stored in each of the GN nodes respectively. The $m$Select algorithm was proposed to select the $m$GN nodes in the network, based on network connectivity and GN node deployment densities. The proposed algorithm ensured that the set of $m$GN nodes selected is the smallest required, so as to reduce the overhead incurred on the network due to a large number of operating mGN nodes. The attack detection scheme also had a pattern update phase, during which individual subpatterns for each of the $r$ target nodes of the network are updated to depict accurate energy content values of the target nodes in terms of the numbers of traffic packets receivable by them in a given epoch of time.

A tradeoff equation was formulated to compute the optimal length of a time epoch, $\Delta_{opt}$, for the scheme to converge in, so as to achieve reasonable attack detection rates at the cost of minimal energy resource usage by the detector and the mGN nodes. The purpose of this formulation was to accommodate the needs of varying application of the network. Certain applications of wireless sensor networks require the scheme to converge less frequently so
as to reduce the overhead, and increase the lifetimes of the network, at the cost of lower attack detection rates. On the other hand, other applications require the scheme to converge more frequently to increase the attack detection rates, at the cost of more resource usage. We incorporate both scenarios within the tradeoff formulation for computation of the optimal time window length.

In Chapter 5, we studied the performance of our proposed distributed attack detection scheme, for variations in several algorithmic and network-level parameters, namely, network traffic intensities (adversarial nodes) and node deployment densities. The experimental results acquired for analysis and comparisons, were quantified in terms of the following metrics:

- Attack detection rates.
- False positive rates.
- False negative rates.
- Node energy decay rates.

As part of the scheme evaluation, we analysed the attack detection rates for variations in the total number of detector nodes in the network, as well as the network sizes and node deployment densities. Subsequently, we performed simulation experiments to study the false positive and false negative rates of the scheme. The attack detection rates showed a significant increase with increasing numbers of attack detector nodes in the network. For smaller node deployment densities, the detection rates were lower, even when very few packets (both attack and normal) penetrated the network. The reason for this degraded performance was inferred to be the unavailability of detector
nodes in several regions of the network, thus leading to the reconstruction of incomplete patterns of observed network traffic. Therefore, attack decision making was not completely accurate in such scenarios. We inferred from the study on pattern update rates that the need for having dynamic pattern value updates within the detector nodes is essential to achieve high detection rates. The performance of the scheme was significantly affected in the absence of a pattern update process in the detector nodes.

The false alarm rates of the detection scheme were compared for various node deployment densities and traffic intensities. The inability of the detector nodes to reconstruct entire patterns of traffic observations in low node density networks, led to higher false alarm rates. Higher accuracies in pattern reconstruction yielded fewer false alarm rates for larger values of $N$. We can therefore conjecture that to increase the accuracies in attack detection, higher node deployment densities are essential.

We also performed experiments to study the energy decay rates depicting the rate of decline of energy content in the detector/mGN nodes. The results showed that the energy decline rate is significantly higher in the mGN nodes, as compared to the GN nodes. The added tasks of receiving attack decision packets, and further analysis, and forwarding of messages to the base station, affects the energy decay rates of the mGN nodes. Therefore, it was concluded that it is essential to have fewer mGN nodes operational in the network, as part of the detection process, to increase network longevity. The proposed $m$Select algorithm in Section 4.4, thus aided in achieving this optimality for defining the total number of mGN nodes for the detection scheme.

In the second phase of experiments, we benchmarked the simulation results for the distributed attack detection scheme with corresponding results
obtained using a centralised SOM-based detection technique. The comparison yielded significant performance improvements of the distributed detection scheme, over the SOM-based approach. The lack of a pattern update mechanism in place for the latter led to reduced detection rates, and increased false alarm rates, with reducing target node lifetimes. The need for a pattern update mechanism in sensor network applications, demands a corresponding update mechanism in place in the corresponding attack detection scheme. The SOM-based approach is applicable for network applications requiring no update in the pattern values post-initialisation.

In Chapter 6, we presented a cluster-based, node compromise/failure-tolerant mechanism for detection of distributed denial of service attacks. The set of attack detector nodes of the network are clustered into an optimal cluster size, defined as $c_{opt}$. The optimal cluster sizes were computed based on a tradeoff equation, formulated to optimise between the attack detection rates and the overhead associated with the scheme in terms of energy usage. The energy usage associated with higher numbers of clusters was less for larger $c_{opt}$ values. Smaller values of $c_{opt}$ increased the detector node energy usage.

The clustered attack detection scheme was designed to operate as an overlay on an underlying network topology, or as a routing topology in itself, wherein the cluster-based approach for data delivery to the base station must be adhered to. Simulation experiments were performed to test the effectiveness of our scheme for varying algorithmic and network-level parameters. We studied the energy decay rates, false alarm rates and attack detection rates of the scheme for variations in the traffic intensities, compromise node ratio and $c_{opt}$ values.
We performed a comparison of the single-cluster approach (Detection scheme from Chapter 4) with the proposed multi-cluster scheme, in the presence of compromised nodes. The multi-cluster scheme performed significantly well, thus implying that sensor network applications vulnerable to node compromise, i.e. in the presence of compromised nodes, there exists a need for having multiple clusters operational in the network, to achieve higher success in attack detection, when a distributed pattern recognition approach for attack detection is used.

7.2 Future Work

The set of colluding adversarial nodes participating in the attack may opt to send requests to the target nodes at regular intervals of time by staying well below the attack detection threshold. In other words, the intensity of attack traffic may be classified by the GN nodes as normal, where in reality, the traffic is constituted of malicious packets intending to cause damage to target nodes over a longer period of time. This type of an attack will lead to a gradual decline in resources of the target nodes. We can refer to this attack as a slow poisoning attack. A future direction of work can involve detection of such attacks in addition to detection of high traffic intensity attacks, addressed in this thesis.

The proposed attack detection scheme does detection of attacks, that culminate from higher orders of incoming traffic within a single time epoch, without correlating traffic behaviour from previous time epochs. This work can be extended to incorporate correlation between time epochs, for attack detection purposes. In addition, the length of the time epoch, $\Delta_{opt}$, is static.
post-initialisation. Variable time epoch lengths, based on analysis of real-time
network traffic, is another possible future direction of research.

The proposed attack detection scheme requires a set of detector nodes to
be operational in the network. The detector node ratio, and its impact on
the success rate were studied through experiments. The results given in this
thesis span the entire spectrum of possibilities for this particular parameter.
However, there exists a need for defining an optimal set of detector nodes,
selected based on the topological placement of the detector nodes, the network
taxonomy, as well as other criteria, such as density of node deployment, to
ascertain a desired level of success in attack detection. Such a parameter can
be fixed at network initialisation time based on the type of application of the
sensor network.

The proposed detection scheme relies on the collective decision-making of
a set of mGN nodes in the network. Another significant future contribution
can be the proposal of a probabilistic decision-making approach, wherein a
randomly selected subset of the mGN nodes will participate in the attack
detection process during any given time epoch. Such an approach can yield
reasonable detection rates, at the cost of lesser energy resource utilisations
by the detector/mGN nodes. Another possible extension to this work is for
the mGN nodes to coordinate with each other, and reconstruct subpatterns
at a second layer of the scheme. Such an approach can help avoid a single
point of failure, for each set of GN nodes, in a given mGN node’s jurisdiction,
that is present in the current detection scheme. As a result, the overall false
alarm rates of the detection scheme can also be reduced.
References


**URL**: [http://theregister.co.uk/content/1/12394.html](http://theregister.co.uk/content/1/12394.html) Verified on: 17 May, 07


247


